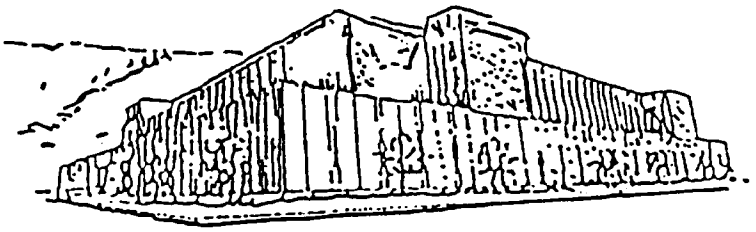
1996

Silicon poetics| The computer as author and artifice

Lawrence Andrew Koch
The University of Montana



Maureen and Mike
MANSFIELD LIBRARY

The University of **MONTANA**

** *Please check "Yes" or "No" and provide signature* **

Yes, I grant permission ✓

No, I do not grant permission _____

Author's Signature Maureen E. Hall

Date May 3, 1996

SILICON POETICS: THE COMPUTER AS AUTHOR AND ARTIFICE

by

Lawrence Andrew Koch

B. A., Rutgers College, Rutgers University, 1991

Presented in partial fulfillment of the requirements

for the degree of

Master of Arts in English (Literature option)

The University of Montana

1996

Approved by:

Chair, Board of Examiners

Dean, Graduate School

5-3-96

Date

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Koch, Lawrence Andrew, M. A., May 1996 English

Silicon Poetics: The Computer as Author and Artifice (115 pp.)

Director: Veronica Stewart

This thesis explores how various computer programs construct poems and addresses the way several critics respond to these computer generated texts. Surprisingly, little attention has heretofore been paid to these programs. Critics who have given the matter attention usually focus on only one of the myriad programs available, and more often than not, such scholarship concludes with a disparagement of all such projects. My work reexamines computer generated poetry on a larger scale than previously exists, positing some conclusions about how these texts affect contemporary theories of authorship and poetic meaning.

My first chapter explicates the historical debate over the use and limits of technology in the generation of text, studying similitudes between certain artistic movements and computer poetry. This historical background reveals that the concept of mechanically generated text is nothing new. My second chapter delineates how the two main families of computer poetry programs actually create these texts. Computer programs combine existing input text, aleatory functions, and semantic catalogues, which provides insight into how humans both create and interact with these programs. At the same time, this study illustrates the difficulty in defining the level of intention and influence by individuals on the textual product, and therefore these texts challenge our traditional notions of authorship and the value of poetry. My third and final chapter argues that contemporary literary theory and poetics creates the conditons under which computer generated poetry can pose as a human product. The success of these programs to deceive readers about the origins of the text becomes clearer with the results of a survey I conducted in which the respondents were fooled by the machine more often than not. This possibility of machine-created text masquerading as human art threatens many critics, who quickly dismiss the process and its results as non-poetic, but I conclude that since the computer complicates foreknowledge of origin in some contemporary poetic forms, this intrusion by the machine prompts us to reconsider how we traditionally value and interpret poetry.

Table of Contents

Introduction

> But the amazing growth of our techniques, the adaptability and precision they have attained, the ideas and habits they are creating, make it a certainty that profound changes are impending in the ancient craft of the Beautiful. In all the arts there is a physical component which can no longer be considered or treated as it used to be, which cannot remain unaffected by our modern knowledge and power.
> --Paul Valéry, "The Conquest of Ubiquity"

In "A Defence of Poetry," Percy Shelley writes that "[p]oetry, in a general sense, may be defined to be 'the expression of the imagination'" (956). Publishing in 1840, Shelley argues that poetry exists as an exalted human product, noting that it "in a more restricted sense expresses those arrangements of language, and especially metrical language, which are created by that imperial faculty whose throne is curtained within the invisible nature of man" (Shelley 958). At about the same time and in another part of England, Charles Babbage was working on a machine that would eventually be known as his "Analytical Engine," and which the world would widely regard as the first programmable computer, operating with a decimal rather than the modern binary system. Although it never ran, Babbage's Engine used a location for a set of instructions (much like the cards of a Jacquard loom) by which the machine would perform its functions. It also allowed for data to be called from one location, manipulated, and stored in a different place. Of course, today's electronic, digital computers operate at speeds millions of times faster

than the Engine could have run, but Babbage's machine, although fettered by the limits of mid-nineteenth century technology, combined abilities never before built into any machine. Able to operate from a program and manipulate data outside of human interference, it was a crucial step towards a technology that might one day, as Babbage hoped, substitute "machinery, not merely for the skill of the human hand, but for the relief of the human intellect" (Kurzweil 165).

One hundred and fifty years later, both Babbage's machine and Shelley's view of poetry have evolved to a unique point in human history: the successful programming of a machine to create what some might call poetry.[1] Computer scientists, literary theorists, and poets have been programming computers to compose text since the 1960s, but only recently have realized successful results. The fruition of these projects emerges not only from advancements in computer technology but also from experiments in several poetic schools that create the environment through which the output of a computer poetry program can mimic its human counterparts with some modicum of success.

With their immense speed and storage capabilities, computers use brute force to compensate for the inability to "think" in the same manner as the human brain. They can

work in hostile environments of temperature and atmosphere, control indefatigable industrial robots that function more precisely than human laborers, and calculate complex problems faster that any person, but no computer has been able to master that which allows us to define, express, and explore our "humanness"--our language. Nevertheless, language engages in an undeniable relationship with technology. Literary art reflects advancements in technological evolution, often upsetting traditions and expectations in favor of new forms of expression, and these cycles of change appear in the area of poetics as well. For example, the introduction of written languages to supplant oral literatures allowed for a text to be arranged as visual phenomenon; manifest and tactile, marks on the page could be sized and ordered by the author for specific effects. In the fifteenth century, Gutenberg's moveable type permitted for more efficient production and distribution of written texts, and in turn, created a greater audience for these works. The production of the printed word vastly increased with the invention of the portable typewriter, which placed typographical symbols of the printing press on the desk at the immediate disposal of the authors. Historically, these advancements in technology generate changes in poetic form--e.g., pattern poetry in the 1600s, the typewriter work of e. e. cummings, and the concrete poetry of John Hollander--and

these new methods subsequently affect the way we conceive and interpret poetry. Most writers today view the computer simply as an efficient and sophisticated typewriter, where word-processing programs manipulate textual elements much more cleanly and easily than an old portable Royal or Olivetti.

But the separation between the computer and the pen or typewriter emerges in the computer's ability to be *programmed*; the computer does not always require human intervention to complete its task, aside from an initial instruction set. This removal of a human subject who can be directly connected with the output complicates the assignation of authorship; many critics contend that the programmer is ultimately the author of these texts, yet upon closer investigation we find that this distinction becomes difficult to maintain.

The presence of the author of a text held privileged status in literature prior to the advent of structural formalism; in fact, the field of literary studies began by primarily focusing on the biography of the author, drawing conclusions and meanings about the function of art by comparing the poet's life to his or her work. But in the first half of the twentieth century, literary critics faced increasing competition from the empirical sciences and, as a result, a new school of literary theory--named New

Criticism--emerged in response to claims of arbitrariness and subjectivity in the field of literary study. This new way of viewing literature separated the author from the work for the first time, arguing that the work of art should be foregrounded as the object of study. For these New Critics, such as W. K. Wimsatt and Monroe C. Beardsley, "the design or intention of the author is neither available nor desirable as a standard for judging literary works," (248) and therefore what the author plans in a work becomes irrelevant and unknowable because the poem "is detached from the author at birth and goes about the world beyond his power to intend about it or control it" (249). Yet despite such claims, Wimsatt and Beardsley still implicitly posit the need for an author, even if lurking about in the shadows, by qualifying that "a poem does not come into existence by accident. The words of a poem...come out of a head not out of a hat" (248-9).

New Critics contended that they could "distinguish between right and wrong readings of a poem" by viewing the work as a structure of norms or standards (Wellek 260), through which the work can only be attributed value and importance as a literary piece according to how well it *achieves* these preconceived standards. The New Critics did not, therefore, completely banish the author. Instead they simply placed his or her importance at a greater remove from

the text--replacing the author with the work as the primary source of meaning.

Poststructuralism, more explicitly the work of Roland Barthes and Michel Foucault in the late 1960s, questioned this repositioning of the author and the perseverance of theories that privilege the author's importance. Barthes, in his essay "The Death of the Author" (1968), notes that the text "is that neuter, that composite, that obliquity into which our subject flees, the black-and-white where all identity is lost, beginning with the body that writes" (54-5). According to Barthes, New Criticism never eradicated the author's power, but "quite often merely consolidated it" (55). In contrast to this methodology, he argues that "[o]nce the Author is distanced, the claim to 'decipher' a text becomes entirely futile" (58). According to Barthes, the text owes its existence to those texts that exist both at the same time and before it; words in the language open up to other words, not towards a final objective meaning: "The space of writing is to be traversed, not pierced; writing constantly posits meaning, but always in order to evaporate it: writing seeks a systematic exemption of meaning" (58).

In a similar fashion, Michel Foucault suggests that "the mark of the writer is reduced to nothing more than the singularity of his absence; he must assume the role of the

dead man in the game of writing" (*What is* 264). Foucault posits this role as the "author function," a position that exists out of a necessity to mark "speech that must be received in a certain mode and that, in a given culture, must receive certain status" (*What is* 267). This "author function" does not correspond to a particular individual, but rather arises from the act of recognizing the work as literature; it emerges as "the result of a complex operation which constructs a certain rational being that we call 'author'" (*What is* 269).

Even in Foucault, this "rational being" implies that the text must have a human producer. In 1969, computer generated text was in its infancy, but today's programs appear able, in some circumstances, to be mistaken for human authors, especially because poetry changed drastically since Shelley's "Defence." Ezra Pound, Gertrude Stein, T.S. Eliot and other modernists of the early twentieth century changed the landscape of poetry by reconceiving poetic form at the same time that the work of the Dadaist and Surrealist movements challenged the traditional relationship between art and artist. New Criticism and poststructuralism, by reducing the importance of the author to varying degrees, initiated new forms of poetry that engaged these new theories. The projective verse of the Black Mountain school in the 1950s supposedly recreated the breathing of the poet,

an idea that implies an author in some relationship to the text. In reaction, the Language poets, influenced by poststructuralism and the Dadaists, undermined this idea of "voice"--the speaking subject--by attempting to fracture language. Their abolition of the authoritative "I" of the poem relied on the concept of language as a system of signification, an intricate play of words that can be manipulated and subverted to reveal power and oppression as it exists in the historically real world.

This shift by some poets towards "meaningless" language, or at least towards texts that do not overtly claim to conform to traditional conceptions of the poem as a vehicle for meaning, coincided with yet another theoretical movement away from the text itself towards audience. Transactional, speech act, and reader-response (or reception) theories all posit the reader as the *producer* of meaning, and this focus--combined with poetries that react against form and meaning--provides the opportunity for computer generated poetry to exist. Computer poetry can successfully mimic these poetries, because the ego created by the human "I" no longer needs to be present, creating a location in which to consider those texts generated by machines as art. But the absence of the author complicates our classification of computer texts as such, because we may not always be able to determine whether or not a machine

created the poem. Even if a human poet claims complete authorship for a text, a skeptical critic will guard against becoming the victim of a human poet's pranks or intentionally misleading comments.

As readers and critics, we face a new challenge about how to view poetry, primarily because New Critical methodologies still influence academic pedagogy as well as composition of anthologies today. If we view poetry as an evolving, rather than stagnant art form, as a process that includes structural experimentation, then we have to be just as flexible in our consideration of the agency responsible for these new forms.

* * *

In the chapters that follow, this thesis illustrates how various computer programs construct texts and discusses the way several critics respond to these computer generated texts. Surprisingly, little attention has previously been paid to these programs. Critics usually focus on only one of the myriad programs available, and more often than not, such scholarship concludes with a disparagement of all such projects. I consider this work as an effort both to reexamine computer generated poetry on a larger scale than presently exists and to posit some conclusions about how these texts affect contemporary theories of authorship and poetic meaning.

By manipulating textual data at blinding speeds and removing human subjectivity through random operations, the computer relies on two inspirational--and controversial--time-honored features of textual production. The mechanical production of text and the decision to relegate the compositional process to chance have occupied authors for over three hundred years, and my first chapter aligns computer generated poetry with these two traditions. An explication of the historical debate over the use and limits of technology in the generation of text and a study of the similarities between certain artistic movements and computer poetry, reveals that the concept of mechanically or electronically generated text is nothing new. Instead, the output of computer poetry programs today, and the promise of more complex and sophisticated ones in the future, appears as an initially successful realization of this tradition of mechanical textual generation.

My second chapter explores how the two main families of computer poetry programs actually create these texts. Although there exists two definable categories of text generating programs, this taxonomy includes programs which vary in such a degree that this classification can only be considered general at best. But how computer programs combine existing input text, aleatory functions, and semantic catalogues provides insight into how humans both

create and interact with these programs yet simultaneously illustrates the difficulty in defining the level of intention and influence by individuals.

These texts challenge our traditional notions of authorship and the value of poetry and my third chapter delineates how contemporary literary theory and poetics allows for computer generated poetry to pose as a human product. The success of these programs becomes clearer with the results of a survey I conducted in which the respondents were deceived by the machine more often than not. This possibility of machine created text masquerading as human art threatens many critics, who quickly dismiss the process as non-poetic. But under closer inspection, however, it appears difficult to maintain these claims because central to most disparagements is the foreknowledge that a text was generated by a machine. Yet, the process through which any author creates a poem often remains unknown to the reader, and certain critical theories reveal the present difficulties in attributing intention, with any degree of certainty, to a given author or process.

The modern computer does not "know" what it is doing in the same manner that we may imagine a human poet does, and some argue it never will. But if we can be deceived by a computer's output to the degree that we accept it as a creation of an individual person, then we need to rethink

the ways we conceive poetic theory, the function of the poem, and the role of the poet.

Notes:

1. In a remarkable coincidence, Ada Lovelace, Babbage's associate, linked the fields of computer science and poetry as early as the mid-nineteeth century. Regarded as the world's first computer programmer, Lovelace was also the only legitimate child of George Gordon, Lord Byron--a poet and contemporary of Shelley.

Chapter 1: The Mechanics of Chance

"Perhaps no person can be a poet, or can even enjoy poetry, without a certain unsoundness of mind."
--Thomas Babington Macaulay,
Edinburgh Review 1825

As one version of the old adage goes, "Put enough monkeys in a room with a typewriter and eventually they will come up with *King Lear.*" In actuality, the outcome is more likely to be a broken typewriter than a literary masterpiece, but the search for a feasible, mechanical, non-human system for the generation of texts has a long history, especially in Europe. The scientific discoveries of the sixteenth and seventeenth centuries, along with the Enlightenment period of the eighteenth century, stirred debate concerning the proper applications of empiricism and scientific invention. In 1678, John Peter created mathematical tables by which anyone with only a knowledge of the alphabet and numbers less than ten could create Latin verse. In his work *Artificial Versifying: A New Way to Make Latin Verses*, Peter distributes letters into tables, making extraordinary claims about the capacity of these tables to help essentially illiterate persons write poetry:

> [A]ny one of ordinary capacity, that only knows the A. B. C. and can count 9 (although he understands not one word of Latin, or what verse means) may be plainly taught (and in as little

> time, as this is reading over) how to make thousands of hexameter and pentameter verses which shall be true Latine, true verse, and good sense. (frontispiece)

Peter's poetry by numbers created a stir of controversy in England, and in 1711 Joseph Addison and Richard Steele lampooned his approach (along with similar attempts by others) as misguided and ineffectual. Addressing the "many Artifices and Modes of False Wit," their response to Peter's methodology satirically attacks such applications of technology to the production of wit and intelligence:

> But of all Contractions or Expedients for Wit, I admire that of an ingenious Projector whose book I have seen: This Virtuoso being a Mathematician, has, according to his Taste, thrown the Art of Poetry into a short Problem, and contrived Tables by which any one, without knowing a Word of Grammar or Sense, may, to his great Comfort, be able to compose or rather to erect Latin Verses. (*The Spectator* 356)[1]

Of course, Peter's tables did not turn poetry upside-down, nor did they seem to have much lasting impact on the literary world except as objects of ridicule by the privileged possessors of eighteenth-century wit. Nevertheless, the importance of *Artificial Versifying*

emerges through the implications it raises with regard to the production of texts *mechanically*.

Language is inherently a human production, or at the very least a production of an organic entity, and poetry in the Enlightenment was primarily considered as a preeminent mode of communication. Peter's tables seem far more relevant today because, as we will see, the tabular method predominates one category of computerized text production and manipulation. The author of *The Spectator* article concludes by remarking that the only improvement to Peter's tables would be the construction of a device to automate these tables, one similar to "the project of a *Dutch* Mechanick, *viz.* a Mill to make Verses" (357). The "improvement," of course, is meant as a joke: by using a machine, the Dutch mechanic's method is even more misguided and further removed from the locus of language than Peter's tables. The satire points to the ridiculousness of both Peter's and the mechanic's efforts, emphasizing what the author sees as a movement away from human understanding and communication. While the existence of the Dutch mechanic's mill cannot be substantiated, a fictional version of it appears fifteen years later in the Laputian Academy of Jonathan Swift's *Gulliver's Travels.* At this Academy--a derisive censure of the misdirected application of scientific empiricism and invention--Swift's Gulliver

records the folly of Laputian researchers, including a "Projector" who attempts to revert human excrement back into the original food, an architect who builds houses from the roof down, and a researcher who experiments with ways to make marble soft enough for pillows and to breed a variety of woolless sheep. More important to this discussion, Swift imagines a professor of "speculative Learning" who invents a *word Engine*. Consisting of a massive frame of wires, bits of wood, metal rods, and words written on paper, the Engine represents "a Project for improving speculative Knowledge by practical and mechanical Operations" (156). Operated by forty pupils, the Engine supposedly gives "the World a compleat Body of all Arts and Sciences":

> Every one knew how laborious the usual method is of attaining the Arts and Sciences; whereas by his Contrivance, the most ignorant Person at a reasonable Charge, and with a little bodily Labor, may write Books in Philosophy, Poetry, Politicks, Law, Mathematicks and Theology, without the least Assistance from Genius or Study. (156)

Of course, in the early eighteenth century, only educated white males were privileged enough to engage in the usual "laborious" method, and although Swift read Addison and Steele, Marjorie Nicolson's work points out that the "sources for nearly all the theories of the Laputans [sic]

and Balnibarians are to be found in the work of Swift's contemporary scientists and particularly in the *Philosophical Transactions of the Royal Society*" (112).[2]

The average reader of 1726, with both knowledge of the limits of Enlightenment technology and ideas that realize the human monopoly on language and intent, could appreciate Swift's humorous indictment of the Laputian Academy and modern science. Neither Swift nor his readers could have foreseen the advent of digital computers and how these machines would affect the literary arts. Nevertheless, the operation of Swift's Engine shares remarkable similarities with several poetry generating programs of today:

> The Pupils at his Command took each of them hold of an Iron Handle, whereof there were Forty fixed around the Edges of the Frame; and giving them a sudden Turn, the whole disposition of the Words was entirely changed. He then commanded Six and Thirty of the Lads to read the several Lines softly as they appeared upon the frame, and where they found three or four Words together that might make part of a sentence, they dictated to the four remaining Boys who were Scribes. (156)

Although mechanical, and requiring human power, Swift's machine utilizes a very primitive program: all the words of the language are represented in all forms, although not in

any order; slender wires form connections between words, not unlike the way a semantic catalogue or thesaurus does; and the visual output varies. Although it remains unclear to the reader what function the slender connecting wires perform between words, the output must be left to chance because the pupils must search for short, sensible segments of three or four words--implying that a great deal of the product is nonsense. Likewise, the user of a poetry program may often sort through the computer's output, culling poetic moments from mountains of doggerel and incoherent phrasings.

Theoretically, by manipulating all the characters of the alphabet, including space, punctuation and other typographical characters, one could generate all the possible combinations of the language. In "The Library of Babel," Jorge Luis Borges creates a world where this occurs. The characters of Borges's fictive universe inhabit an infinite, spherical library, and since everything that can be written has been and nothing can be thought of that does not exist somewhere in the library, the inhabitants have nothing to do but wander aimlessly about the corridors overwhelmed by the sheer volume of senseless texts. As with Swift's Engine, groupings of sensible words become artifacts and legendary objects of study: "Another (very much consulted in this area) is a mere labyrinth of letters, but

the next-to-last page says *Oh time thy pyramids*" (53). The narrator responds to the discovery of an understandable linguistic phrase by ascribing its construction to a set of rules that must be contained in yet another undiscovered book. In such an overwhelming creation as this library, permutational computation eliminates chance, and although Swift's scientist never mentions this fanciful and unsettling outcome as a possibility of his machine, contemporary readers of Borges's short story may interpret his library as a future product of a very fast and powerful computer with enough time and memory to complete its task.

The mathematical or mechanical production of language to which all of these authors refer creates problems because such a process can, and usually does, display an incompetency with language. The ability to use a language involves much more than the production of grammatically correct sentences; ordinary linguistic constructions must be semantically correct as well. We often take for granted the knowledge of what makes for sensible discourse because the distinction between nonsense and significance seems immediately obvious to native speakers of a language. But the process of learning to use language is incredibly complex: it involves our sensory experience of the world, it resides in our cultural interactions with others in "real time," it depends on recognition of what Ludwig Wittgenstein

calls the "language game" in which we are currently involved,[3] and it tests our ability to recognize and adapt to neologisms and changing lexical meanings. The computer, like any other machine, lacks this "intelligence" of how to derive signification from linguistic constructions and to use semantically correct language for given situations. Without this knowledge, a computer program relies instead on deception. When asked a question, it produces stock responses that often appear stilted, or it replies with another question--turning the problem back onto the human interrogator.[4] But in creating an output that gives the appearance of linguistic competency, as in the case of poetry, often the computer's program selects textual elements by utilizing a random number generator. Without a semantic understanding of the language, computers alter the results each time and achieve variety by relegating the output to chance operations.

The creation of text through random generation is not the sole domain of mechanical and electronic devices nor is this mode of production alone a valid criteria for dismissing computer generated texts, for chance plays an important role in several modern artistic movements. Although poets in the Romantic tradition viewed the world as order masquerading as chaos, the Dadaists (influenced by Futurism, Symbolism, and Cubism) embraced chance as a new

method of creating art free from the stifling expectations of previous traditions. Whereas chance was previously understood as a subject of poetic discourse, it had now become a necessary principle in the *creation of poetry*. The most famous and prolific artists of the Dada movement--Hans Arp, Tristan Tzara, Hugo Ball, Kurt Schwitters, Andre Breton, and Hans Richter--utilized chance in an attempt to unfetter art from the conscious interference of its creator. Richter, in *DADA: Art and Anti-art*, explains "the central experience of Dada" as follows:

> Dada's propaganda for a total repudiation of art was in itself a factor in the advance of art. Our feeling of freedom from rules, precepts, money and critical praise. . . was a major stimulus. The freedom not to care a damn about anything, the absence of any kind of opportunism, which in any case could have served no purpose, brought us closer to the source of all art, the voice within ourselves. The absence of any ulterior motive enabled us to listen to the voice of the 'Unknown'--and to draw knowledge from the realm of the unknown. (50)

The idea of this unknown realm, supposedly a place where knowledge preexists and one that the artist may harness in the creative process appealed to the Dadaists as a way to

repudiate dominant artistic culture and tradition. Harriet Watts, in her insightful work *Chance: A Perspective on Dada*, remarks that through this unknown "one can posit systems of order, means of cohesion which have not yet been accepted and legitimatized by established intellectual traditions" (155).

In twentieth-century physics and psychology, the unknown became fertile ground for revolutionary ideas, exemplified in quantum mechanics by Werner Heisenberg's celebrated uncertainty principle which posits that simply observing a physical process at the sub-atomic level affects the interpretation of that phenomenon (Kurzweil 116, 193). Therefore, predicting such events becomes impossible, or indeterminant, because of the paradoxically unavoidable interference of the observer. Similarly, Sigmund Freud's work in psychoanalysis had a profound impact on the Dadaists to varying degrees, as Richter confesses:

> Chance appeared to us as a magical procedure by which one could transcend the barriers of causality and of conscious volition, and by which the inner eye and ear became more acute, so that new sequences of thoughts and experiences made their appearance. For us, chance was the 'unconscious mind' that Freud had discovered in 1900. (57)

For the Romantic poets, the concept of the sublime increases perceptual awareness; the Dadaists, on the other hand, view chance as a more effective and objective method of realizing unconscious thoughts and images.

Supposedly, Hans Arp first discovered chance as a poetic instrument when he tore up a drawing in progress and let the pieces fall to the floor. Later, he happened to notice the arrangement of the scraps on the floor and realized that the pattern they formed revealed a quality he had been unable to attain consciously. Richter explains Arp's amazement by noting that "chance movements of the hand and of the fluttering scraps of paper had achieved what all his efforts had failed to achieve, namely *expression*" (51). Interpreting the genesis of this 'expression' proves more difficult than recognizing it; Richter cannot decide whether it was "the artist's unconscious mind, or a power outside him, that had spoken? Was it a mysterious 'collaborator' at work, a power in which one could place one's trust? Was it a part of oneself, or a combination of factors quite beyond anyone's control?" (Richter 51). While the Dada movement embraced Arp's *Das Gesetz des Zufalls*, or "Law of Chance," individual artists of the period, in pursuit of the answers to these questions, employed chance differently in their respective works.

For example, Tristan Tzara also experimented with automatic writing and generated random texts using newspaper articles as the impetus for his art. His "Dada Manifesto on Feeble Love and Bitter Love" outlines his most famous experiment:

> TO MAKE A DADAIST POEM
>
> Take a newspaper.
>
> Take some scissors.
>
> Choose from this paper an article of the length you want to make your poem.
>
> Cut out the article.
>
> Next carefully cut out each of the words that makes up this article and put them all in a bag.
>
> Shake gently.
>
> Next take out each cutting one after the other.
>
> Copy conscientiously in the order in which they left the bag.
>
> The poem will resemble you.
>
> And there you are--an infinitely original author of charming sensibility, even though unappreciated by the vulgar herd. (Tzara 39)

Despite their similar methodologies, Tzara and Arp each viewed chance with a slightly different perception,

and Watts illustrates their differing methodologies by considering Tzara's process. Apparently composed from a mutilated newspaper article shaken out of a paper bag, Tzara's first line reads "when dogs cross the air in a diamond like ideas and the appendix of the meninx tells the time of the alarm programme (the title is mine),"[5] to which Watts observes:

> The most striking feature of the text is the fact that all attempts to impose any intellectual order on the flow of verbal elements will be thwarted. Had Arp begun with the [first line], the 'absurd' image would have been developed further. . . . The suggestive range of the image would have been filled out and amplified through Arp's verbal inventions; and the original accident would have been modified by the intervention of the conscious artist. Tzara, however, does not encourage this rounding out of the accidental image. (139)

Tzara's absolute reliance on chance prevents any further interference by the artist because "it is not the way the words fell" (Watts 139). However, this removal of the artist curiously does not preclude authorship, for his manifesto maintains that this random artistic creation or *poem* "will resemble you." Richter turns authorial function into a point of contention:

> Tzara exploited the same chance factors as did Arp, but while Arp made conscious use of his eye and brain to determine the final shape, and thus made it possible to call the work his, Tzara left the task of selection to Nature. He refused the conscious self any part of the process. . . . Arp adhered to (and never abandoned) the idea of balance between conscious and unconscious. This was fundamental to me as well; but Tzara attributed importance exclusively to the Unknown. This was the real dividing-line. (Richter 60)

This "dividing line" between the two artistic positions regarded the intention of the author as determinable, and it separated and placed the perception of the Unknown, or unconscious mind, opposite the conscious mind and viewed these entities as accessible, or at least opened, by chance operations.

In contrast, Andre Breton also experiments with automatic writing, but claims that his method differs from Dada because it functions as a means to explore the unconscious. For Breton, "every product of the mind strove towards conclusion. Dada never concluded" and as early as 1920, Breton broke from the tradition and "attempted to assimilate to Dada his personal ideas about poetry. . . and on the role and destiny of this poetry which to him was

closely related to psychoanalysis"--ideas that would later become instrumental in the Surrealist movement (Hugnet 190). Breton believed the "self" was what elided Richter's idea of the Unknown, and since arbitrary social and cultural conditions exert control over authorial intention, these conditions subsequently affect conscious creation. The insidious tradition of literature (and through implication, Arp and Richter) forces the artist "to correct, to correct *oneself*, to polish, to smooth out, to find fault instead of drawing blindly from subjective treasure" (Watten 44). Automatism, for Breton, allows the artist to break this "slavish custom" of literary production; it denies this "correction" of text to produce a creation free from the subjective control of the conscious mind.

Since automatic writing methods attempt to sever the artistic impulse from the constraints and intervention of the conscious mind, some scholars argue that such work supposedly produces or becomes a testament to the unconscious desires of its creator and therefore provides fertile ground for psychological criticism. According to John Erickson, the automatic poem "while offering an apparent unintelligibility to the reader, does contain certain discursive strategies that are susceptible to analysis" (*Dada: Performance, Poetry, and Art* 79). He argues for the work itself as a product of a certain

process, and that process, ostensibly the endeavor through which the author removes himself from conscious determination, reveals how much the author actually *is* determining the text:

> [Automatic writing] undergoes several mediating processes in becoming words on a page. Far from representing the transcription of raw outflowings of conscious thought, the Dada poem is acted on by several factors: rudimentary thought processes of preconsciousness that impose a manifest form on the latent content of emotion and desire; the interposition of preexisting models (literary texts, media, everyday speech) whose borrowings direct or determine the text; the conscious mind of the poet who modifies the text to achieve specific effects (through lexical substitutions, etc.). (Erickson 79)

The difficulty with Erickson's explication lies in two implicit assumptions: first, he assumes a critical awareness of the text as an outcome of this process; and second, he assumes the author as a split subject with both unconscious and conscious minds. Since the Dadaists, and critics such as Erickson, perceive the work of art as an extension of an individual and therefore influenced by an artist's unconscious, their assumptions rest *a priori* on the

existence of a human author--even though Barthes argues that when "accepting the principle and the experiment of collective writing, Surrealism help[s] desacralize the image of the Author" (Barthes 56). Computer generated poetry undermines these presuppositions of authorial presence; if one attempts to analyze the author of the work, the computer complicates the determinacy of intent, whether conscious or unconscious, because of the difficulty of determining authorship and the randomized, mechanical production of the text. While the second factor excludes a direct relationship between the poetic avant-garde and the generation of texts by a computer, the determinacy of whether an operator of a poetry program alters the output (if one did, how would the reader know?) and the "rudimentary thought processes" of everyone involved--the computer programmer, the operator, and the author(s) of input texts--are both difficult, if not impossible, to ascertain or locate in the output.

Yet, since computers cannot operate without a program written by a human, there is, if only initially, a process of artistic determination made by an individual--but how much we can say about authorship appears uncertain; as I will show later, authorship may often involve *several* authors and random, mechanical intervention. Yet, the creation of text through chance operations necessarily

involves at least one individual initiating the artistic process; as Watts notes:

> The fact that [the artist] has chosen to create in this fashion makes him a causal factor in the whole process; he intervenes, even if only to designate the material that is to be subjected to random influences and to recognize any random object, pattern or event as suited to his own artistic needs. (156)

The question still remains as to how one substantially determines the causal connection and the level of intervention of this individual, and with regard to computer generated texts this may prove ultimately impossible--complicating, at the very least, most previous theoretical approaches in poetics.

Although Dada was proclaimed "dead" in May 1922,[6] the aims of the movement and the concept of chance generation of text was not forgotten. The influential composer John Cage created several musical works and poems through random generation, and since the 1960s, Language[7] poets have experimented with chance in an effort to break free of what Michael Davidson refers to as "bardic, personalist impulses," and explicitly focus instead "on the material of language itself" (*Princeton* 675). By representing language "as such,"[8] stripping words of their assumed, everyday

meanings, the poet creates the appearance of a non-interpretive "self." In such poetry, Barrett Watten concludes, "the 'self' has become generalized as 'language,' or, put another way, the 'self' has exploded and disappeared. . . The mediating persona has been abandoned" (Watten 52).

Jackson Mac Low, a contemporary Language poet, generates texts through which, as Charles Bernstein suggests, "the language is exteriorized, no longer a transparent transport to a given world depicted (Bernstein 252). Often, his poetry comes with performance instructions, allowing the reader to co-author the production of the text. For example, "Is That Wool Hat My Hat" consists of four separate voices to be read by up to four individuals, led by a "conductor" who keeps a beat. In addition, Mac Low's "diastic" technique, an linear acrostic method somewhat similar to Cage's mesostitch, creates poetic texts that appear to be generated randomly, when in fact their composition follows a specific formula, e.g., his work "Ridiculous in Piccadilly" where he composes eleven poems from Virginia Woolf's *The Waves*. Mac Low explains his method this way:

> After finding the title phrase. . . I drew one word for each of its letters. Beginning with the

> phrase itself, I culled only words in which the letters occupied corresponding positions (I disregarded hyphens): "*r*idiculous P*i*ccadilly.//en*d* sta*i*n/book*c*ase,/reass*u*ring brutal*l*y/eating-h*o*use.//eating-h*o*use.//waitresse*s*,/*i*n a*n*d *p*lates r*i*ght/inc*l*uded.//pri*c*k cont*a*ins forge*d* compan*i*on/pale-ye*l*l*o*w/smooth-po*l*ished melancho*l*y/" upon which, having spelled the phrase out once, I began again. ("The Genesis of 'Ridiculous in Piccadilly'")

The systematic method through which Mac Low constructs these texts "refuses the normal process of identification of a 'self' (voice, persona, sensibility) *in* the text as expressed or revealed" (Bernstein 252). To generate poetry in this manner, either through the chance rolls of a die or through a systematic template which appears random, involves an extensive time commitment on the part of the author; a commitment that can be reduced to a few seconds with the aid of a computer. Outside of an input text and an initial set of parameters, a computer can produce poetry in the same manner as Mac Low, precisely because Mac Low's methods are, to a considerable degree, mechanical and reproducible.[9]

* * *

The artistic legacy of the Dadaists, combined with historical attempts at the mechanical generation or manipulation of texts, provides part of the poetic framework for computer generated poetry; it is a tradition through which the randomization of textual data offers a new way of perceiving both language and the world. "For creative and rebellious minds of the late 19th and 20th centuries," as Harriet Watts suggests, "Western aesthetics and rationalism were either totally discredited, or insufficient to render experience in its totality," and "chance was an obvious phenomenon to which to turn as a keyhole to the unknown, where other possibilities might well exist" (155). This discourse, shaped historically by repeated attempts of poets to break out of conventional modes of artistic representation, ultimately allows a context in which language rises to the "surface" of a poetic text, confounding an anticipation of meaning or significance. If the output of computer programs can masquerade as poetry, one reason is because contemporary poetic movements have opened the door for these texts to appear as such.

In such a tradition, language becomes concretized, "visible on the page, sounding at the level of each phoneme, so that the phonemes turning to morphemes turning to words turning to phrases turning to 'poem' is felt, heard, made tangible, palpable" (Bernstein 70). The disdain of Language

poets for "symbolic language and hidden layers of meaning" (Conte 274) produces a new, object-oriented process, through which "there is no separation of fact from language, no layering of discourse, no transport from one plane of existence to another--only the intricate play of/at the surface of language itself" (Conte 275).[10]

Since experiments in computer generated poetry reveal the current limitations of computer technology to produce texts that rely on subjective, encoded meaning (such as the narrative or thematic elements that occur in epic poetry), consideration of computer texts as poetry takes place both through a comparison of these texts to human products and the reader's recognition of certain "poetic" elements--a method which considers the text as object. Often the program, attempting to generate a balance between both traditional and the avant-garde forms, results in a "poem that is nothing but 'an object in and by itself' even though the programmer is trying to make the poem look not like such an object but like a traditional poem" (Newell 168). Using the failure, or more accurately the inability, of the computer to create a text imbued with discursive strata or signification as a dismissal of any poetic value in the text is only a result of certain established modes of interpretation; situating the output in both Modernist and contemporary traditions, I argue, allows the text a position

as poetry. How individual programs utilize the two historic traditions of mechanical and chance production of texts differs from program to program; but to mimic the variegated results of the human creative process, all computer poetry programs must rely on the random manipulation of textual data. The next chapter explores this random process through the different taxonomies of computer poetry programs and their outputs, and examines how poetry by a machine can, in certain instances, confound our notions of authorship.

Notes:

1. The satire of Addison and Steele's *Spectator* article relies heavily on the predominant codes of the literary hegemony at that time, the legacy of which still exists today in several facets of the production of literature, including academic canon formation, creative writing programs, and awards committee--all of which assign or maintain "literariness" in poetic works. The "author" of the article also satirizes those authors who assign merit by how quickly they finish a work, reaffirming the notion that art must be painstakingly produced over a lengthy period of gestation; and another who never prints his work but inscribes poems on the window glass of taverns with a diamond ring, and ceases writing forever when he loses his "Genius and his Ring to a Sharper at play" (*Spectator* 356). Of course, a computer can create text faster than any human, and could be considered as a tool similar to the diamond ring, and these reasons have been used to disparage computer poetry. Even though modern experiments in poetry and form allow the computer's output to pose as a human production, it appears that the theories of some critics have not changed in almost four hundred years.

2. Many critics disparage "Voyage to Laputa" as trite and ineffectual compared to the other travels of Gulliver, and several authorities, including W. A. Eddy, note that there

"seems to be no motive for the story beyond a pointless and not too artfully contrived satire on mathematicians," and defer to assign its creation to "one of Swift's literary ideocyncracies [sic]" (Nicholson 110). Yet, Marjorie Nicholson suggests that this is not the case:

> The very fact that the literary and political background of Gulliver's Travels has been established so completely leads the inquisitive reader to inquire whether the unrecognizable sources of the Voyage to Laputa may not be equally capable of verification. If the most assiduous searcher into sources can find 'no literary source or analogue' for the peculiar themes in this voyage alone, must not those sources be sought elsewhere than in the literary traditions Swift inherited? (*Science and Imagination* 112)

3. In fact, Wittgenstein opposes the view that language can be reduced to explicit rules and formulas:

> For remember that in general we don't use language according to strict rules--it hasn't been taught us by means of strict rules, either. We, in our discussions on the other hand, constantly compare language with a calculus proceeding to exact rules.

> This is a very one-sided way of looking at language. In practice we rarely use language as such a calculus. For not only do we not think of the rules of usage--of definitions, etc.--while using language, but when we are asked to give such rules, in most cases we aren't able to do so. We are unable clearly to circumscribe the concepts we use; not because we don't know their real definition, but because there is no real 'definition' to them. To suppose that there must be would be like supposing that whenever children play with a ball they play a game according to strict rules. (*The Blue and the Brown Books* 25)

4. ELIZA and PARRY are two examples of such programs. Eliza, written by Joseph Weizenbaum in 1966, simulates a nondirective (i.e., Rogerian) psychotherapist. Ray Kurzweil, in *The Age of Intelligent Machines*, cites a commonly quoted example of its output:

> Subject: Men are all alike.
>
> ELIZA: IN WHAT WAYS
>
> Subject: They're always bugging us about something or other.
>
> ELIZA: CAN YOU THINK OF A SPECIFIC EXAMPLE
>
> Subject: Well, my boyfriend made me come here.

ELIZA: YOUR BOYFRIEND MADE YOU COME HERE

Subject: He says I'm depressed much of the time.

ELIZA: I AM SORRY TO HEAR THAT YOU ARE DEPRESSED

Subject: It's true. I'm unhappy.

ELIZA: DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY... (Kurzweil 204)

Similarly, Kenneth Colby's program PARRY attempts to simulate a paranoid patient who believes the mafia is out to get him. When a group of psychiatrists were given a domain-specific Turing test (See Note 2.1), which involved PARRY and a real, live human suffering from paranoia, the psychiatrists did a little better than chance at distinguishing between the two (Kurzweil 54-5). These two examples are given to show the limits of a computer's "understanding" of language--in these cases, it consists of a program that formulates replies based on prompts in the input text.

5. The full text of Tzara's poem appears as a note after "How to Make a Dadaist Poem" as an example of his process. One must remember that not all Dadaist poems were created this way, yet nevertheless it shares remarkable affinities to several computer generated poems discussed in Chapter Two regarding semantic and syntactical confusion in the output. Tzara's poem reads:

> when dogs cross the air in a diamond like ideas and the appendix of the meninx tells the time of the alarm programme (the title is mine) prices they are yesterday suitable next pictures/appreciate the dream era of the eyes/pompously that to recite the gospel sort darkens/group apotheosis imagine said he fatality power of colours/carved flies (in the theatre) flabbergasted reality a delight/spectator all to effort of the no more 10 to 12/during divagination twirls descends pressure/render some mad single-file flesh on a monstrous crushing stage/celebrate but their adherents in steps on put my nacreous/sumptuous of land bananas sustained illuminate/joy ask together almost/of has the a such that the invoked visions/some sings latter laughs/exits situation disappears describes she 25 dance bows/dissimulated the whole of it isn't was/magnificent ascent has the band better light whose lavishness stage music-halls me/reappears following instant moves live/business he didn't has lent/manner words come these people. (Tzara 39)

6. With typical predilection for overblown performance, the figureheads of the Dadaist movement held a formal ceremony

to lay Dada to rest: "In May 1922 Tzara, Van Doesburg, Arp, Schwitters, and Richter held a funeral service for Dada at the Bauhaus festival in Weimar. Dada, as a concerted activity, had ceased" (Erickson 119).

7. Throughout this paper, I use the more contemporary term "Language" rather than the more awkward and dated spelling of "L=A=N=G=U=A=G=E."

8. An embodiment of traditional poetic discourse encountering the work of either Dada or Language poetry can be found in Wittgenstein's observation that "The confusions which occupy us arise when language is like an engine idling, not when it is doing work" (*Philosophical Investigations* 132).

9. In a personal letter to a fellow poet regarding the creation of mistakes in his work *Words nd Ends from Ez*, Mac Low responds:

> The making of mistakes is the true intervention of unsystematic chance, and as such one might think that I ought to accept it, or even welcome it. I'm of at least three minds about it. "Reading-through methods" are not chance operations even though they are nonintentional insofar as I cannot know ahead of time or determine consciously what words, etc., will enter the texts produced by means of them. In principle they are

> "deterministic" (as my astrophysicist son puts it) even though they are intentionally nonintentional in the above sense.
>
> But determinism is fucked up by the mind and its lapses and by my never having time enough to check and recheck them. So be it. It is better (and more "Buddhist") to accept them after a certain period of time. (Letter to Karin Schalm, 1994)

10. What may appear as a single layer of discourse in fact consists of many intertextual relations, notably that of poetry theory, programming theory, and the individual contributions of programmer and operator. This is discussed in more detail in the next chapter. For Conte, Language poets produce texts that appear as non-intertextual entities and in one sense, this attempt succeeds--although this implicates theories of reading and whether meaning in a poem lies in the text or in the reader.

Chapter 2: The Generation of Text

"Poetry is what Milton saw when he went blind."
--Don Marquis, quoted in
E. Anthony, *O Rare Don Marquis*

In the early 1960s, researchers in artificial intelligence began to experiment with programs that could parse sentences and create phrases, with the goal of eventually producing a computer capable of understanding human speech and writing. Ultimately, such a program would have command of the language to a degree that it would be able to answer questions from a human interrogator in a variety of subjects, and deceive the human judge into believing that the computer was an actual human being. Known as the Turing test,[1] this exam has become the holy grail for AI researchers, and some even speculate that a computer will pass this test as early as 2020 (Kurzweil 483). Known as natural language programming, this area of computer science dealing with language recognition and understanding was the initial birthplace for computer generated poetry programs. Although these early programs produced semantically stilted language and their form was limited to correct sentences, they impressed their programmers by occasionally creating coherent text. Most of these initial experiments were limited to haiku generating

programs, and often were products of more "serious" research projects.

But in the late 1970s and early 1980s, the introduction of the desktop personal computer produced a wave of dilettante programming hobbyists able to compose, mostly in BASIC, a wide variety of applications in their living rooms. Before this time, access to and production of these programs was confined to those who could afford the space and cost of large mainframes--mostly large research corporations and universities. With the new technology of home computers, novel attempts were made to manipulate and produce texts, and out of these efforts, several rudimentary poetry generating programs emerged. But the limits of this new technology, including slow processing speed, limited language applications, and inadequate storage space and memory, soon became apparent and most forays into natural language processing in the home were stalled, abandoned, or dismissed. In the past fifteen years, personal computers have become so powerful so quickly that a new interest in the computer production of text now appears to be emerging. Yet, even with new advances in hardware and software, natural language programming and artificial intelligence have still not produced a computer able to "know" what words represent (outside of 1s and 0s) in the way that any human does. Therefore, even the latest programs today utilize one

of two systems of generation that were well-implemented by the mid-1980s. These two systems, as the last chapter illustrates, have an established tradition in the arts, and both incorporate an aleatory function to produce output. Louis T. Milic, one of the early optimists of computer generated poetry, labels these two different systems as *formulary* and *derivative* (*Princeton* 230).

The primary difference between these two methods lies in the source of textual elements. Formulaic systems operate with lists of words, often grouped by parts of speech and predetermined by the programmer or operator, while derivative programs rearrange existing input texts according to certain sequences of probability. Sometimes known as "text manglers," derivative programs allow the possibility of combining Shakespeare's sonnets with Pound's cantos. As simple as these two classifications sound, the application of each method varies greatly within this taxonomy, necessitating a closer analysis of each method and its proponents. Since these differences become crucial to understanding the complexity of "authorship" in computer generated poetry, I will compare a few selected programs from each case and present their output for consideration.

Formulaic programs combine words according to templates established by the programmer; these templates appear as the equations that govern sentence construction. One template

might appear as simple as "Noun+Verb" or as complicated as "Article+Noun+Verb+Preposition+Article+Noun," the latter of which might create "The cat sits on the rock" just as easily as "A cars eats at the tree." More sophisticated programs use templates that designate tense and number in an attempt to eliminate the semantic disagreement of the second result, while some may even use a rudimentary thesaurus to select parts of speech appropriate to the subject--milk, dog, or kitten, but not blue or cloud might accompany or modify "cat"--but it does not require much consideration to realize how this could severely limit the output while still producing errors. To be effective, the staggering number of semantic connections necessary would have to rival the network of the human brain's ten billion neurons. Considering that each of these neurons could have up to 200,000 separate entry ports, or connections to other neurons, configuring an artificial brain for language lies far beyond today's technological boundaries (*Gödel* 340).[2]

One of the most famous formulaic programs, *Racter* (short for raconteur) by Bill Chamberlain and Thomas Etter, claims to incorporate a primitive thesaurus, while also earning recognition as one of the oldest of the second generation of poetry programs that began in the early 1980s. In 1984, Chamberlain compiled *Racter*'s output into *The Policeman's Beard is Half-Constructed,* a text that

spurred a flurry of anticipation about the computer's potential to create prose and poetry. In his preface, Chamberlain claims: "With the exception of this introduction, the writing in this book was all done by a computer," and that "the programmer is removed to a very great extent from the specific form of the system's output. This output is no longer of a preprogrammed form. Rather, the computer forms output on its own" (Chamberlain [introduction]).

Following such assertions, the reader encounters such seemingly intelligent work as:

> A hot and torrid bloom which
> Fans wise flames and begs to be
> Redeemed by forces black and strong
> Will now oppose my naked will
> And force me into regions of despair.
> (Chamberlain 7)

Chamberlain points out that his program has the ability to assign a "variable" status to certain randomly chosen words and use them at key points in the text; the resulting output of this process appears:

> to spin a thread of what might initially pass for coherent thinking throughout the computer-generated copy so that once the program is run, its output is not only new and unknowable, it is

> apparently thoughtful. It is crazy "thinking," I grant you, but "thinking" that is expressed in perfect English. (Chamberlain [Introduction])

Computer programmer Jorn Barger, on the other hand, indicates that "only the most generous interpretation of these claims will hold up under close scrutiny" (Barger). What appears as coherent syntax, Barger claims correctly, exists because of prewritten templates in which *Racter* only substitutes a small number of elements. In fact, many sentences or lines in some examples do not contain any variable elements, leading Barger to conclude that the style of the text belongs not to *Racter*, but to Chamberlain.[3] Unfortunately, this information, coupled with the popularity of *Racter,* has either led to an inaccurate and widespread disparaging of all computer-generated texts or to the quick assumption that sole authorship of the output belongs to the programmer.

In *Poetry CreatOR*, Erik Sincoff modifies *Racter's* approach by introducing more randomization and prompting the user for certain elements such as title, subject, and a synonym. Rather than utilize one template for the entire text, Sincoff's program consists of many possible templates for each line, grouped by lines one through nine, yet chosen at random from within those groups. Between one and four variables exist in each line, and each variable further

limits the selection field by defining the subdivision of each category. Sincoff arranges verbs, for example, by violence, movement, noises, thinking, or temper, and adjectives by color, emotion, speed, or shape.

Once running, the program selects the first line from a data list of possible first line templates such as:

"$VERB3$!" $VERB2t$ the $ADJ1$ man, the keeper of $SUBJp$

The numbers after the variables indicate which group the program chooses from: three and two represent thinking and noise verbs respectively. In my experiment, I chose to have the program select a subject at random, (although I could have input one of my own) which produced: "Remember!" spoke the agitated man, the keeper of monkeys. The following text represents a full sample output:

> The arguing pair fantasize with a sorrowful ear
> Stalking nothing like it was yesterday--just like a cello.
> Ever briskly, the leg took its toll. . .
> Eating wolves as snack food, the lurking mass blocked the sun
> My favorite thing is typing "Marie Antoinette is a Virgin" on your depressed face
> Never remembered. . . never more has been argued
> Hope for you is not a morning dove, rather a mourning dictator

From a height high above, the preacher grasped the orange.

O melancholy world, you have operated me again.

Although Sincoff's program only produces nine lines of output, each of those lines may have up to four variable elements allowing *Poetry CreatOR* to accomplish more than *Racter*. Sincoff's program improves upon *Racter*-style programming by increasing randomization, and although there still exists "echoes" of human influence, the technique increases the distance between the programmer and the actual output.

The appropriately titled *DadaPoem Generator* by Alex Chachanashvili represents one of the most simple applications of formulaic programming, yet provides the most distance between the programmer and the text because it fully randomizes each discreet unit of text. Beginning with a data list of templates, the program "fills in the blanks" with data from lists of nouns, verbs, adjectives, adverbs, conjunctions, and prepositions. Once started, the program selects a template such as %A_%ving_%n_%vs_%J_%n%x and begins to insert the proper information, where "a" represents an adverb, "v" a verb, "n" a noun, "j" an adjective, and "x" a random punctuation mark. Capital letters indicate that the article "a" or "an" should precede the selected term, while an underscore represents a space

and the percent sign separates terms in the sequence. The program only recognizes the first term after the percent symbol as a search string, adding any additional characters which follow. For example, the first verb in the above template would appear in the present tense with an "ing" ending. The whole line yields, in actual output: "a beautifully thinking poet sees a soft woman!" With carriage returns as a possible template choice, the program may randomly introduce blank lines which the operator might use to group the output into stanzas. Yet, while always grammatically correct, the output rarely appears semantically coherent. From hundreds of lines of output, one of the more sensible groupings appears as:

An angry writer remorselessly writes
About an old glass
A poem deliberately mangles shoes.

Walking--
A woman sweeps the blue and violet skies
Dissonantly felt, because people listen sexually
A verdant devil abruptly drives over loneliness.
An angry watch
Mirrors a city, and further
A decayed feeling about mirrors sings
A thought that rarely comes on poems.[4]

Even though illusory, the serendipitous appearance of "writer" with the appropriate verb "writes" in the first line combines with the line breaks to mimic traditional human elements of language and poetic construction. Random selection produces the former, while templates of partial sentences create the appearance of enjambed text. With a word-processing program, I modified the vocabulary data lists, omitting such words as "brain tissue" and "monster truck" in favor of traditional, and perhaps hackneyed, poetic terms such as "thought," "heart," and "emotion." My conclusion: abstract nouns seem less constricted in association than concrete ones, producing more semantically plausible constructions.

One final example of formulaic programming produces output that combines the coherence of *Racter* with the degree of randomization *DadaPoems* offers.[5] Created by Chris Westbury, *McPoet* uses templates with built-in randomization of their elements so that the templates themselves change each time the program selects them. In order to accomplish its humorous title screen boast of "Doing for poetry what McDonald's has done for food," *McPoet* randomizes the appearance of certain elements by dictating the probability of that element appearing in the template. For example, one template reads: 100 "do" 100 "not" 90 "simply" 10 p_verbs_from 100 macroReturnSpacing 50 adjectives 50 p_nouns

100 punctuate 70. The numbers following each variable indicate the percentage chance of that unit appearing in the output. "Do" will appear 100% of the time, while "not" will follow 90% of the time, and 10% of all runs of this template will contain the modifier "simply." Both the subject and verb, which agree in number, will appear every time although the actual nouns and verbs substituted will be randomly chosen each time. Interestingly, of the four programs mentioned, only *McPoet* randomly manipulates the outward form of the text by implementing a random line break, in this case, the template will output as one line only 50% of the time, and every other instance as two. *McPoet* also allows the operator to determine the number of lines, the subject, and the subject's gender. This ability to influence the output directly becomes crucial when one realizes that the program does not accommodate neuter subjects, only masculine and feminine. Therefore, to appear coherent, the operator must use names, occupations, or objects with conventional genderings. (Ronald Reagan as male, earth and nature as female, and poet or author as either.) Such varying probability often produces mixed results, one line insightfully reads, "Derrida loves awhile, he understands absolutely and ritually/his pert fad" while, with "poet" as feminine subject, the program produces the comical:

A poet splashes,

she buzzes,
many big deadly cents excavate.
Her big black old baboon is like the chaste nubile
    extroverted delicacy,
it sings to us,
her song is like innumerable offensive
adorations.

Only three lines were requested at the start; one can easily see that *McPoet*'s ability to produce random line breaks adds the illusion of human intention, although the content of the text itself appears absurd.

Formulaic poetry generating programs produce texts influenced by two individuals: the programmer and the operator. One could argue that they are one in the same, since by inputting data such as subject and gender, the operator enters into the role of programmer and "finishes" the instruction set. It would follow that in such a case, the label "programmer" now applies to a role and not to a specific individual. Much to the possible disappointment of the Bill Chamberlains and Chris Westburys of the programming world, authorship now disintegrates into a true author "function," not applicable to identifiable individuals. Yet somehow this creates a nagging sense of inaccuracy precisely because of the *type* of language computer programmers use.

We may put the term author in quotes when we refer to composer of *Paradise Lost* or *Leaves of Grass*, because the language of those works differs substantially from the language used to create computer programs. Languages such as BASIC and LISP evolved out of symbolic mathematical language in use since the 17th century. In natural (human) language, the sign may be arbitrary and ambiguous as many theorists posit, but in formal (computer or symbolic) language the sign explicitly represents an operation. Therefore the intent of the author in a section of programming code is always obvious, so that interpretation of the sign never necessitates interaction with culture or society. Ferdinand de Saussure was one of the first linguists to realize this; in his *Course in General Linguistics*, Saussure points out that "one characteristic of the symbol is that it is never wholly arbitrary; it is not empty, for there is the rudiment of a natural bond between the signifier and the signified" (Saussure 73). Unless modified by someone else, the symbolic language (the program) of *McPoet* always exists as Chris Westbury's, yet in interacting with Westbury's program the operator creates a situation not where Westbury's contribution cannot be determined, but rather, the operator now becomes *co-programmer* with the original author.

This relationship in the co-authoring of the program

creates a level of textual discourse that lies beneath and preexists the actual output of the program. This discourse-below-discourse most certainly affects both how the computer will create its "poem" and what this work will contain--similar to the manner in which human poetic discourse and theory influences the way human poets compose their work. Likewise, although the computer itself may not experience an "anxiety of influence" in its production of text, the programmer attempts to create a program whose output emulates the form of existing poetries--indicating some degree of a relationship between poetic and programming discourses.

Yet in these formulaic programs, the inclusion of a random generator complicates a co-programmer relationship at the level of output. The programmer creates an instruction set and vocabulary data lists, the operator sets parameters and may supply additional elements, but the machine itself manipulates the output randomly--three distinct, yet sometimes individually undefinable, contributions to any given output. Computer generating programs of the other classification, the derivative variety, further complicate this tripartite arrangement because they produce exactly what their name implies: text derived from other texts.

Most derivative programs work in much the same way that

Tzara's paper bag does, yet they incorporate methods of analysis that help reduce the number of incompatible phrasings. Famed literary critic Hugh Kenner, along with Joseph O'Rourke, created one of the most famous derivative programs in 1984. Named *Travesty,* Kenner and Rourke's program reads an input text according to a "Order" that the operator assigns:

> If the order is 4, it seeks out every occurrence of the initial 4-character sequence and records the character that comes after. It then chooses at random from its notepad one character to append, moves forward one place, [and] repeats the whole process. (Hartman 77)

Charles Hartman and Kenner used *Travesty,* along with Hartman's *Diastext,* to produce a work called *Sentences* in 1995, in which the input text contains 457 nineteenth-century grammar examples from the Thayer Street Grammar School (Providence, RI) composed by Samuel Green and known as "Sentences for Analysis and Parsing" (Hartman 79). After reading the 3,250 word text, consisting of such examples as the title itself, "School begins," "Dogs barked," and "John will forget," *Travesty* produces an initial output according to the order chosen. One such output appears like this:

> Sentences for Analysis and Parsing Thayer Street Grammar School begins. James, bring me the vessel

had been using that that. Our little lame. He hurricane. The love of money is to prepare forsaken. Iron has brought it tremble. The young must do it is. (Hartman 7)

Rather than accept this output as final, Kenner and Hartman input this output into another program modeled on Jackson Mac Low's diastic technique of verse composition. *Diastext,* authored as mentioned above by Hartman, composes works in the same way as Mac Low creates "Ridiculous in Piccadilly." Using the above output from *Travesty,* Hartman's program systematically produces:

Sentences begins.

money must
Sentences
Parsing
Sentences
Sentences
Sentences for love forsaken.

Analysis and
Thayer
Analysis money
Analysis
Analysis

Analysis and

Analysis and

Parsing

James,

Our forsaken. (ll. 1-17)

Diastext continues until it has reached the end of *Travesty*'s output. Kenner posits that this final text demands to be read aloud; the repetition produces a rhythm which "implies Chant, therefore Voice" and he notes "a setting for massed choirs and soloists is not inconceivable" (Hartman 82). Interestingly, the end result evolves from several initial sources: Hartman's selection of Green's hundred-year-old list, Kenner and O'Rourke's *Travesty,* and Hartman's *Diastext.* Green's list becomes the output of *Travesty,* which in turn becomes the source text for *Diastext,* and the reader finds the final output bearing very little resemblance to the original.

In a somewhat similar manner, Stefan Strack's *Mark V. Shaney* reads an input text or texts and builds a "word probability table" that reflects the probability of a given word following a sequence of words (Strack). The program initially views the text according to the "word grain"--which represents the number of words to be read as a unit. If the word grain is "2," the program first builds a table of all the words in the text by groups of two: the first and

second words, the second and third words, the third and fourth words, etc. Once completed, the program then calculates the probability of a third word following each two-word group, stores that information, and chooses a word cluster at random--producing text according to highest probability, stopping only when it reaches the last word of the input text. With a word grain of "2" the output closely resembles the input; change the word grain to "1" and the text becomes fully, yet systematically, mangled from a random starting point. Since the program recognizes punctuation as part of the word, the product appears remarkably grammatical.

The same method of probabilistic determination drives Neil Rubenking's *Brekdown*,[6] inspired by *Travesty* and similar to *Mark V. Shaney* except that it works on the level of the character rather than the word. Using *Brekdown*, John Tranter creates texts derived from the work of both Matthew Arnold and John Ashberry (Tranter). After reading a continuous input text from Arnold's "The Buried Life," "Dover Beach," and "The Scholar-Gypsy" *Brekdown* produces output from which Tranter culls several lines. Of the thirty-four lines he presents, one notices a striking level of coherency not previously attained with other programs:

> From the dying pastoral slopes an unwanted earth art
> gone

> And the vast edges draw back the impulse of an
> hour--
> Exhausted, thou waitest for one desire, and the soft
> Abstractions of reapers in the intellectual trough.
> So wild brother men, concealed then with distracted
> air--
> Let it be spent on other joy, and we,
> Wanderer one of antique shadow, rest
> And in the bluebell-drenched days, men
> Who in the sun, thy fire their being roll.
> Come shepard, bathe in our war of antique shadow,
> 'Tis this story of the wooden bridge, wrapt in
> disguise. (ll. 19-29)

Using two anagrams of "Matthew Arnold," Tranter titles the work "What Mortal End" and attributes authorship to "Tom Haltwarden." While the jumbled nature of the anagram may aptly represent the process of composition, to what degree does this output text belong to Matthew Arnold? The apparent fluidity of the output can be deceptive, for Tranter himself indicates that he edits the output, gathering "the thirty or so 'best lines' of that raw text" and "clean[ing] them up a bit to make them less garbled" (Tranter).

Whereas the question of authorship becomes troubling enough in formulaic programming, it becomes even more so in

a derivative process. The above example combines the written program of Rubenking, the diction and structure as created by three of Arnold's poems, and the editing skills of Tranter. And much like a co-authored human work where each individual's contribution may be difficult to discern, one can easily imagine how indistinct each individual's contribution might become in derivative programming, especially if the input text comprises the work of more than one author--does that appearance of "science" refer to Milton's use or Poe's?

So why apply the fictitious name of one person to Tranter's example rather than title it "Output #1" by "Neil Rubenking, Matthew Arnold, and John Tranter"? Besides raising the issue of how, in 1992, John Tranter and Neil Rubenking can co-author a text with a man dead for over a hundred years, this process implicates the question of ownership in a capitalistic society. Supposing this text becomes marketable as poetry, would royalties be paid to everyone involved--the operator Tranter, the programmer Rubenking (who may be unaware that his program composed such a text), and the owner of the rights to Arnold's work, a poet who, since he died in 1888 most certainly remains unaware of his contribution?

Although Arnold may not be the best example in this case because of American copyright law, what if we replaced

Arnold with a poet still alive, or only dead for a few years? In light of the way we often attribute ownership to human authors, this question of who we credit with authorship may seem rhetorical to some. Traditionally, we discount the texts of multiple authors, such as the "exquisite corpse" projects of the Dadaists, because of the method of production and the difficulty in determining each individual's intent and contribution. Curiously, it seems that the way that "group" poetries are composed results in their output being treated as non-serious or non-intentional, much like computer poetry--so long as we know how they were produced. These texts thwart attempts to define authorial intent because the work is not seen as a unified whole; it is not a product of one individual's efforts which contains the voice and intent of that individual.[7] Even in "found" poetry (where the author finds the poem as text preexisting on such mundane objects as newspapers, parking tickets, or cereal boxes) or texts that contain highly intertextual material, we tend to assign authorship to the individual who claims responsibility for "finding" these texts and assembling them in a certain way. But in computer poetry, the process by which one "finds" this poetry consists of selecting the input texts and sifting through mountains of output and *deciding* what will make it off the screen into print--and each operator's ideas

of what a poem is and what it should look like obviously influence this decision.

So, in accordance with accepted convention, having an imaginary author or anagram attached to Tranter's example allows the text to attempt to conceal its origins and present itself as a possible human construction. Admittedly, as readers encountering a text with an unrecognizable name, we usually default to the assumption that it is a human construction--but can one distinguish the computer's output from the human works it imitates?

Using output from his *Kurzweil Computer Poet,* Raymond Kurzweil composed a domain-specific Turing Test to see to what extent people could detect computer texts from the work of human poets. After selecting stanzas from four human poets (Percy Shelley, T. S. Eliot, William Carlos Williams, and Kurzweil himself), Kurzweil produced output using an input text created from poems by these same authors and asked participants to judge whether they thought a computer or a human composed each stanza. Some of the more easily recognizable human contributions, such as Eliot's "I should have been a pair of ragged claws/scuttling across the floors of silent seas" appear interspersed with the computer's "O thou,/Who moved among some fierce Maenad, even noise and blue/Between the bones sang, scattered and the silent seas." (Lines produced by mangling the work of Eliot and Shelley.)

Although his survey group was small, including only thirteen adults and three children, and does not indicate whether any of the judges were professional poetry readers or scholars, the results indicate that the judges were correct, on average, only 63% of the time (Kurzweil 377). Considering that the simple rule of chance between two choices produces at least a 50% probability, it appears the computer stanzas often deceived the judges successfully.

If this program truly succeeds in producing texts which can deceive readers into believing that some human authored them, then it appears that the way we have traditionally viewed the relation between poetry and authorship needs to be reconsidered. But before we assert such a claim, we need a larger study of these texts and the human responses to them, preferably with a survey that does not include children and focuses on respondents familiar with poetry and poetic form. Such a study also needs to include output from more than one computer generating program, in order to test the field as a whole, and to determine which programs have a greater success rate.

As we have seen, the programs mentioned in this chapter vary widely both in their methods and output, and to devise an accurate method of testing a poem for the nature of its authorship will most likely prove impossible, given the human tradition of subverting genre and form distinctions.

In other words, what would prevent a human poet from imitating computer output? With the evolving sophistication of the programs available today, and those available in the future, it appears increasing evident that computer generated poetry can no longer be dismissed as a passing fad.

Notes:

1. The Turing test derives its name and methodology from Alan Turing, a mathematician and computer specialist who, in 1950, wrote "Computing Machinery and Intelligence" in which he proposed the following test of a computer's intelligence (Kurzweil 48). In one room, a human judge sits at a terminal connected to two other terminals in one or more rooms. At one of the two terminals is a human, the other is connected to a computer, and the human judge must determine by the answers to a series of questions via teletype which respondent is the computer and which is the human. This test considers the computer to be "intelligent" if it succesfully stumps the judge.

Since the judge can ask any question he or she wants, including questions designed to elicit responses concerning emotions, feelings, and even ask jokes, etc., it comes as no suprise that no computer has yet passed this test. But researchers have successfully used "domain-specific" versions of this test to establish a computer's expertise in highly focused and specific areas, such as answering math questions or imitating paranoid schizophrenics (see note 1.4).

2. Even if one could devise such a language map, AI philosopher Hubert Dreyfus notes that the ability to understand and learn language necessitates intelligence as

an "embodied" form (Dreyfus 255). An artificial language map would appear as a set of rules for usage, yet Dreyfus argues that our associations to and meanings within language arise from having the sensory input of our physical bodies, not from a rigid rule book. This implies that for any AI computer to be able to "understand" and use language correctly, it must have senses and mobility equal to that of a human being--it must be able to move about and experience the world, not sit on a desk on in a laboratory. Similarly Wittgenstein also maintains that the idea of an artificial catalogue runs opposite to our everyday experience of language. (See Note 1.3)

3. Barger quotes the following example to illustrate his point. The actual Policeman's Beard 'output':

> At all events my own essays and dissertations about love and its endless pain and perpetual pleasure will be known and understood by all of you who read this and talk or sing or chant about it to your worried friends or nervous enemies. Love is the question and the subject of this essay. We will commence with a question: does steak love lettuce? This question is implacably hard and inevitably difficult to answer. Here is a question: does an electron love a proton, or does it love a neutron? Here is a question: does

> a man love a woman or, to be specific and to be precise, does Bill love Diane? The interesting and critical response to this question is: no! He is obsessed and infatuated with her. He is loony and crazy about her. That is not the love of steak and lettuce, of electron and proton and neutron. This dissertation will show that the love of a man and a woman is not the love of steak and lettuce. Love is interesting to me and fascinating to you but it is painful to Bill and Diane. That is love! (Chamberlain [3])

According to Barger, the actual underlying template appears as:

> Key:
> <text variables>
> (redundancies explicitly added by Chamberlain, by repeating a text-variable type, apparently for camouflage)
>
> <Intro phrase> my own (essays) about love and its (endless) pain and pleasure will be (understood) by all of you who read this and (talk) about it to your (<worried> <friends>). Love is the (subject) of this <essay>. We will <begin> with a question: does <meat> love <vegetable>? This question is

> (<implacably> <hard>) to answer. (Here is a question: does a man love a woman or, (to be specific), does <man> love woman>?) The (interesting) response to this question is: no! (He is (infatuated) with her.) That is not the love (of <meat> and <vegetable>). This <essay> will show that the love of a man and a woman is not the love of <meat> and <vegetable>. Love is (interesting) to me and you but it is painful to <man> and <woman>. That is love! (Barger 1)

4. I have edited this by capitalizing the first letter of each line and by correcting "feeled" to "felt" in line three.

5. Several other generating programs exist, most notably those formulaic varieties that produce haiku verse. In the late 1960s, Margaret Masterman and Robin McKinnon Wood generated the short, three-line verse form on a computer at the Cambridge Language Research Unit. Using a simple schema, it constructed: "All green in the leaves/I smell dark pools in the trees/crash the moon has fled." Two decades later, programs such as Haiku Master by Andrew Stone still create text in approximately the same fashion: "[T]he inner sun/attempts summer stillness./O cloistered ineffable!" Because these programs exclusively compose in haiku verse form, and have not evolved noticeably since

their conception, I have not attended to this subdivision here.

6. Although I was unable to procure a copy of Brekdown--not even Rubenking himself has a copy of the program--I sampled a similar shareware program entitled Babble! by Tracey Siesser, Lee Horowitz, and Jim Korenthal, which appears to produce output in a similar fashion.

7. A large number of textbooks and other "literal" or nonfiction works contain several authors, often listed and explicitly claiming multiple authorship. (Indeed, many "autobiographies" of several celebrities utilize the talents of professional writers.) But it is interesting to note that fiction and poetry (because they are seen as *more intentional*, perhaps?) almost exclusively contain single authors.

Chapter III: Postformalism or Deception?: Discerning the Author/Reader Relationship in Computer Generated Poetry

> "A poem need not have a meaning and like most things in nature often does not have."
> --Wallace Stevens, "Adagia"

As computer generated poetry programs become more sophisticated, their impact on the poetry community may become difficult to discern or measure. For both poets and academics, the ability of the computer to produce poetic text raises several issues dealing primarily with the need to define what constitutes "poetry." Raymond Kurzweil's small survey mentioned in the last chapter pitted his text-mangling program, *Kurweil Computer Poet,* against the works of T. S. Eliot, Percy Bysshe Shelley, William Carlos Williams, and his own non-computer verse. His respondents answered correctly little more than half the time--better than pure guessing, but still not a convincing win for human poets.

In fact, Kurzweil's study should not be extrapolated to conclusions about all computer poetry programs; rather, a more accurate study of computer generated poetry needs to take into account a larger portion of the field of available programs. Additionally, as noted in the last chapter, Kurzweil's study of only sixteen individuals included three children, and several respondents replied that they did not have much experience with poetry. A computer may appear

capable of deceiving those with limited exposure or interest in poetry, but could the same program or others accomplish similar results if the individuals surveyed possessed a high degree of poetic knowledge?

In an attempt to determine the actual quality (what some might call "usefulness") of computer generated poetry, I composed a survey of poems (or stanzas if the poem appeared too large to quote in full) composed by both human poets and the programs I illustrated in the previous chapter.[1] Of the fifteen examples, eight were composed by either *Haiku Master, DadaPoem Generator, Travesty, Kurzweil Computer Poet, McPoet, Poetry Generator,* or *Brekdown.* The human poets ranged from Language poets Ted Greenwald, Clark Coolidge, and Jackson Mac Low to more well-known, anthologized poets such as T. S. Eliot and Amiri Baraka.

A total of thirty-seven students and professors at the University of Montana responded, and since all currently study literature or creative writing, I expected results far less encouraging for the computer than Kurzweil's survey received. Since none of my respondents are children and most replied that they read poetry occasionally or most of the time, I anticipated that a large percentage of these students, poets, and academics would immediately discern the sheep's clothing of the computer texts. But the results showed quite the opposite, with the entire field of

respondents answering correctly only 46.2% of the time and the highest individual number of correct answers tallying eleven of fifteen.[2] Does this domain-specific version of the Turing Test prove that the output of a computer program appears indistinguishable from that of a human poet? Before trying to answer that question, we first need to review the methodology I applied to construct my survey.

Obviously, the results would differ if I had selected some of Shakespeare's sonnets and contrasted them with nonsensical output from a text-mangling program. But since the responding field was comprised by students and teachers of literature, and the object of the survey was to seek conclusions based on analysis of the textual elements in each example, the examples themselves could not contain easily recognizable work of any given author. Such examples test an individual's memory, not their analytical ability. But, in fairness, the human examples came from humans recognized as poets, rather than from dubious sources. Two factors influenced the process through which I selected human works from collections or anthologies. First, since chance plays such a large role in computer output, it follows that the longer the output, the greater the opportunity for the computer to reveal its random processes and for the text to lose the illusion it attempts to create. Also, what I perceived as limits on the size of the

survey, both in copying costs and respondents' interest levels, precluded the quotation of large blocks of text.

The second factor arises from semantic problems. Since most individuals have a preconceived conception of the random nature of the computer program, they may assume that any text constructed in a syntactically and semantically correct way must have been composed by a human. But since the metaphoric nature of poetry allows for a reader to understand or interpret a semantically awkward phrase from a human poet, an individual may base his or her answer not on how much sense a poem contains, but how much sense can be attributed to it. With these two factors on mind I applied something of a role reversal: I chose those human poems that might appear to be computer ones, and chose computer poems that resembled human output, resulting in an "average"--examples that could easily appear as one or the other.

If one considers that a blindfolded respondent has a 50% chance of guessing the correct answer, then those surveyed answered correctly at *less than chance*--indicating that the individuals surveyed established their decision on a set of criteria to which each example either matched or did not. For this criteria, two possible positions exist: one, the idea of how a computer poem might appear, or two, the idea of what a human poem should look like. Since none of the individuals surveyed answered all the questions

correctly, one can assume that whatever the reader establishes as criteria must be subjective and not objective. Amazingly, T. S. Eliot--one of the more famous poets in our language--was mistaken for a computer by almost one-third of respondents. Combine this with four out of five people assuming the output of *McPoet* as a human construction, and one can see present difficulty in determining where some modern poetry might come from.

Yet, this difficulty in recognizing human authors from computer ones may not appear important if the reader approaches the text in a certain way, because the results of this localized survey indict certain methods of interpreting poetry. Faced with the challenge that computer poetry offers to those theories that primarily focus on the author, whether biographical, psychoanalytical, or historical, critics have disparaged computer generated texts. The most fervent protest comes from those arguing for the communicative nature of an author-reader relationship and from critics wishing to preserve a "high culture" perception of the art. Considering that poetry has long "been considered a wasteful thing for an able-bodied man to do, when he could be earning a living in a more serious way" as Louis T. Milic notes, the reaction to computers infiltrating what some consider the domain of humans seems slightly ironic (Milic 169).

Nevertheless, in "Speculative Equations: Poems, Poets, Computers" Howard Nemerov introduces his discussion of computer generated poetry by questioning whether or not computers can even write poetry, and suggests that such a question appears to "indeed look new. . . as well as faintly silly" (Nemerov 394). If anything should strike one as amusing in his naive dismissal of the machine, it appears to be his opinion of the nature and utility of poetry:

> Supposing it to be technically possible to make a computer write what will technically pass for poetry, we have still to ask about the poetry it grinds out with such frightening industry and at such tremendous speed whether it expresses the soul of the computer or the soul of the programmer. (Nemerov 395)

Nemerov derives this notion--that poetry should only reflect the "soul" of its author--from the classical debate between the poetics of Plato and Aristotle, wherein he summarizes the question of morality as an integral component of composition. Nemerov argues, quite presumptively, that all of "our ideas of what poetry does and how it does it" can be aligned on either side of this debate. He notes that, for Plato, the poet must be outside himself, possessed by a muse or god which dictates the words, while for Aristotle, the poet "is a rational and conscious craftsman," where

conscious thought guides his work (400). According to Nemerov, the computer appears to be aligned on the side of Plato, "where the poet is regarded as oracular, vatic, not speaking so much as spoken through by something other than himself" (400). Yet, one cannot adduce much from this association, as Nemerov proceeds to liken the computer to an "Other," again as a muse or god. At the end of his rambling essay, Nemerov offers three reasons why he hopes that poetry (which fits his model of form and content) will remain outside the abilities of computer scientists. He bases his first explanation on the apparent readership of poetry:

> In a world where practically no one reads poetry, it is not really desirable, and may not even be sane, to increase exponentially the number of objects called poems, thus giving some poor idiot the task of deciding whether in fact they are or aren't. (Nemerov 412)

His abusive rationale might be easily dismissed as humorous if it were not for his claim that these reasons *not* be considered "frivolously intended" (412). It becomes difficult to take his comments seriously without contesting his insulting assumption that the respondents of my survey, including professors and graduate students, are nothing but "idiots."

Nemerov continues his rationale against computer generated poetry by ludicrously positing the idea that writing poetry involves excitement and pleasure, as opposed to the insane creation of text via a machine: "The dullness, the want of gaiety and charm in the idea, are as appalling as the absence of sanity" (412). He also notes that "[t]he advantage claimed for the computer is its immense speed, but programming it, on the other hand, looks to be a slow and laborious and rather uninteresting business" (412). I imagine a great number of computer science professionals would disagree with his opinion of computer programming as a "rather uninteresting business." And while the "immense speed" of the computer certainly appears as an important feature, Nemerov overlooks that this speed must also be combined with randomization and information storage to generate text.

Nemerov's last reason consists of two questions: "Why should the idea ever have come up at all? To what need in the human spirit does it respond?" (Nemerov 413). In the first chapter, I pointed out that mechanical manipulation of text has a long history, and it appears sensible that one would apply these efforts to the digital computer--the latest evolution in machine technology. The second question Nemerov poses infers that computer poetry should satiate a

"need," although he never offers an explanation of what desire human poetry supposedly fulfills.

Concluding his argument, Nemerov speculates on a future outcome in the relationship between computers and poetry:

> [I]f poetry did come to be written by computers and people read and even declared they loved that poetry, one would still have to suspect that what happened was not so much that the machine had imitated the subtlety of the mind, but that the mind had simplified (and brutalized) itself in obeisance to its idol the machine. (Nemerov 414)

That the mind could be guilty of self-flagellation and idolatry to the machine seems to me to necessitate an initial separation of the two, something difficult to determine in our Western culture at the end of the twentieth century.[3] It appears that, to Nemerov, the pastoral poetries of Shelly or Sir Philip Sidney showcase the majesty of the human mind because they adhere to a traditional, established form, while the less "poetic," and experimental, work of Louis Zukofsky and even Ezra Pound represents the reader suppressing or compromising his or her mental faculties--another difficult distinction to maintain in the face of Contemporary Poetry studies.

This notion of "obeisance to its idol the machine" also seems important for Josef Ernst, whose article "Computer Poetry: An Act of Disinterested Communication" attempts to "analyze computer poetry as an artistically adequate, although misguided, representation of the structure of the postindustrial 'information society'" (Ernst 451). Apparently computers should know their place and remain relegated to cash registers and video games because, like Nemerov, Ernst finds it disturbing "that the act of thumping the keyboard erases the ambiguities of language cherished by traditional literature," and bemoans the idea as "not a problem of computer technology, but of its application" (Ernst 452). But does this really happen? Computers produce ambiguities in language through the randomization of the textual elements programs draw from, and preferencing a "natural" human quality of ambiguity while regarding computer output as unambiguous somehow implies, once again, the ability to differentiate between the two.

The conclusions Ernst draws concerning computer poetry focus solely on the output texts of William Chamberlain's *Racter* program, which as we have seen, contains the least interference by a machine and would be the least likely poster child for a serious study of computer generated text. In order to deny the worth of computer poetry, Ernst posits language as subordinate to politics and notes that the need

to communicate "is inherent in every human activity, it is ultimately the political interests behind individual output that shape the substantive discourses" (Ernst 454). To illustrate his point, he quotes this haiku-like *Racter* output:

> In a half bright sky
> An insect wraps and winds
> A chain, a thread, a cable
> Around the sphere of water

With the assumption that individual output infers *human* output, he concludes that *Racter* "neither initiates a conscientious critical process. . . nor is it based on a recognizable human interest" and therefore "[w]hat looks like a poem and reads like a poem is not a poem" (Ernst 455).

Yet his rationale, without answering the begged question of what a poem supposedly "looks like" or "reads like," appears contradictory: "The typographic pattern on the page and the highly subjective use of language--arranged in an old-fashioned grammatical and syntactical order--makes the above piece identifiable as poetry" (Ernst 455). If one of the criteria for defining a poem is the use of "old-fashioned" (I assume he means "correct") grammar and syntax, then Ernst seems to be at odds with most of what has been produced as poetry in this century.

In fact, if the text was announced as a poem, the reader could construct a reading by focusing on these grammatical constructions. For example, the "half bright" sky creates a difficult image to visualize; the sky does not appear cloudy, but rather it lacks exactly half of some degree of illumination that could be defined as "bright" and may suggest mystery and ambiguity. In the second line, the subject of the text appears as an insect--one might imagine a spider, but the text does not specify a type of insect or even if the insect produces, or simply manipulates, the "thread" of line three. The insect "wraps and winds"--the juxtaposition of which contrasts the two differing connotations. Wrapping insinuates the idea of packaging, clothing, or hiding something, while winding infers the storing of the filament for the future, much like one winds wire or thread.

But the third line, containing the terms describing this filament, may perhaps be the most interesting and fruitful grammatical construction for interpretation. The three terms "chain," "thread," and "cable" appear linearly in the reading process, so that the next term in the sequence modifies the previous ones. Chains are strong, binding, oppressive, and connote possession, while a thread appears as the opposite. The delicate nature of thread joins objects more tenuously and less decisively than chain,

but here "cable" appears in a sort of Hegelian dialectic of the two previous terms, a synthesis or average in which the last term read rests privileged in the reader's mind. The cable, thin as a thread while maintaining the strength of a chain, strikes a balance between the two terms. The indecisiveness of the metaphoric third line intensifies the ambiguity of what the "sphere of water" consists of, and the difficulties involved in such a task. This exegesis could lead to a number of final interpretations, including the analogy of the insect to a satellite in space. Circling in the "half-bright" sky of the earth's diurnal rotation, the satellite--appearing as an insect both in its insignificant size and in its physical resemblance with antennas and solar panel wings--"wraps and winds" both its orbit and its electronic signals around the Earth, a planetary sphere mostly composed of water. The invisible electronic cable of television, radio, or other telecommunication appears as delicate as a thread, but yet our modern world's reliance on these signals forms an unbreakable chain of dependence.

This short digression illustrates the problem in describing what action I just engaged in. Did I interpret a poem? Ernst claims that it is not a poem simply because of who, or more appropriately what, created it. "Before readers attempt an interpretation of the text," he adds, "they need to interpret their superficial identification of

the piece as a literary genre," noting that the failure of readers to question their criteria of what constitutes an acceptable poem leads to the success of *Racter*'s ruse (Ernst 455). But if I, as reader, interpret those four lines from *Racter* as a poem, can one simply excuse my efforts as empty and insubstantial because I did not have prior knowledge regarding the mode of composition?

All of Ernst's objections seem to lie in our ability to discern human from computer poetry, and according to him the latter cannot be considered poetry because it does not possess the intention to communicate. P. D. Juhl formulates a similar argument, positioning his disagreement with computer poetry in relation to authorial intent. Juhl raises the question "Is the meaning of a poem necessarily the same whether it has been written by a person or produced by chance?" (Juhl 482). Again, the ability to distinguish between human and computer texts appears central, and this introduces a new point regarding authorship. Computer generated poetry implicates the faith we place in the honest responses from humans who claim authorship of given works; after all, if I allege a computer poem to be my own, who would contradict my claim? Since we have not defined criteria by which to differentiate, compositions by a computer may continually problematize any discussion concerning origin of certain contemporary poetries.

Yet, by establishing and maintaining a relationship between speech, meaning, and origin, Juhl posits that "to 'interpret' a computer 'poem' is not to interpret a poem" (481). Most modern philosophy assumes meaning to consist of mutual understanding between two or more individuals, which excludes the possibility of "personal" meaning. Rather, we have personal associations attached to words which others may or may not share, but we constitute meaning socially and culturally. Throughout his article, Juhl appears to use "intention" as a synonym for "meaning," and that confusion leads to the notion that human poems, provided we can distinguish them as such, have a meaning encoded by the author to be decoded by a reader. But if we consider the poem as a vehicle for coded meaning, rather like a cryptograph, then we could argue that only one answer can be the correct one. In this critic's formulation, either you decode the hidden message properly, or you get it wrong.

This somewhat New Critical stance leads Juhl to assume that every aspect of a poem must be considered intentional in order for anyone to be able to say anything about it:

> Clearly, the idea that certain words, lines, or sentences produced by a computer belong together or constitute a whole is unintelligible. In order for us to take certain words, lines, or sentences as belonging together or constituting a whole, we

> must assume that they have been produced by a person and with certain intentions. Thus to call something a poem or even a text is to say, among other things, that the words, phrases, lines or sentences have not been arranged this way by chance but have been produced by a person and with certain intentions. (Juhl 485)

What appears obvious to Juhl actually arises out of a misunderstanding of how a computer composes poetry. He likens it to markings on a rock etched by the wind and asks how one could give meaning to such phenomena.[4] But a computer does not spontaneously compose poetry. Although it contains an inherent random factor, it also requires programming by a human. Template programs have their syntax already defined by the human programmer, and mangling programs work with probability based on an original human composition; therefore both rely on syntactic rules that lie outside of themselves in the human use of language itself--in the same way that humans rely on these external rules. So some degree of intentionality remains in computer programs, but only in the form of base rules for composition.

Juhl's idea of intention has often been aligned with the aspect of poetic voice--the conception of communication traditionally reinforced in the workshop poem. We

customarily refer to the I, or self, in a poem as the "speaker," without often realizing that writing comprises poetry, not speech, and that speech comprises the recitation of poetry. But the tradition privileging the oral qualities of poetry leads to the same privileging of speech over writing, where writing becomes subordinate to or derived from speech. Therefore, only a human self has access to language, which reinforces the assumption that the poet intends to "say" something, a supposition with which Charles Bernstein disagrees:

> It's a mistake, I think, to posit the self as the primary organizing feature of writing. As many others have pointed out, a poem exists in a matrix of social and historical relations that are more significant to the formation of an individual text than the personal qualities of the life or voice of an author. (Perloff 16)

These "social and historical relations" have their foundations in language, in the cultural agreement of meaning. If writing, not the self, lies at the center of poetry, then it suggests that communication and transmission of encoded meaning may be an inaccurate conception of the purpose of poetry and how it operates.

As early as 1971, one of the seminal proponents of computer generated poetry, Louis T. Milic, attempted to devise a better explication of this problem:

> One interesting result of my activity in computer poetry generation is a new definition of poetry. In an important sense, strings of words are interpreted as poetry if they violate two of the usual constraints of prose, logical sequence and semantic distribution categories. . . . In short, since the sentence is obviously well-formed syntactically but does not 'make sense', [sic] it is interpreted as poetry, as part at least of a larger poetic structure. (Milic 169)

In Milic's assumption, poetry becomes wholly figural--the lack of literal "sense" forces the reader to view the text as metaphoric. In the final analysis, however, Milic's conception of meaning entangles him in the same complications as Juhl by arguing that figurative language necessitates a recognition by the reader of a purpose behind the construction:

> We perceive how readily we accept metaphor as an alternative to calling a sentence non-sensical. We always tend, that is, to try to interpret an utterance by making whatever concessions are necessary on the assumption that the writer had

> something in mind of which the utterance is the sign. (Milic 180)

Since the comparison in metaphor never presents itself explicitly, the reader of this construction, to use Donald Davidson's words, becomes "bullied into making this comparison" (Davidson 39). The reader attempts to determine not what the metaphor means (decoding the author's intention), but rather what it *could* mean, and many other factors influence which interpretation the reader ultimately accepts, such as context and a knowledge of the objects compared.

Milic, while supportive of computer generated poetry, views interpretation as futile and empty if no authorial intention exists, concluding that "[i]f we are not to waste our time in vain interpretation we must now ask a new question before beginning an exegesis: Who or what wrote this poem?"[5] (Milic 180). Since he refers to a computer-generated text as a "poem," I do not believe that Milic, as an English professor and proponent of computer poetry, would not recognize the difficulties implied in answering his question. To identify a text as a poem is to recognize certain features that traditionally define certain writing as art, and historically this label has been reserved for human products, which now the computer complicates. Of

course, the possibility exists that Milic asks the question precisely to illustrate this dilemma.

Stanley Fish, in his famous controversial essay, "How to Recognize a Poem When You See One," relates how he deceived his class with an experiment in poetry interpretation. After listing the names of five people whose work he was discussing in a previous class, he told his seventeenth-century religious poetry students that they were looking at a religious poem of that era and asked them to identify it. Using the analytical tools that they had attained in his class and others, they formulated a impressive reading of the poem, an act Fish attributes to one of context: "As soon as my students were aware that it was poetry that they were seeing, they began to look with poetry-seeing eyes, that is, with eyes that saw everything in relation to the properties they knew poems to possess" (Fish 326).

Obviously Fish's "poem" does not contain intention in the way the previous critics I have mentioned attempt to define poetry. Yet, his students thought they were seeing a poem, and acted in a specific manner. Juhl says they did not interpret a poem, and Milic labels that interpretation as vain, but both critics operate from hindsight and additional knowledge, something not always included or accessible to a given reader. Recent reader-response theory

carefully avoids the "anything goes" impression of such an activity, positing instead the idea of interpretive communities which influence the way we read. Based on the reading and interpretation strategies employed by his students, Fish determines that readers create exegeses, instead of finding them:

> Skilled reading is usually thought to be a matter of discerning what is there, but if the example of my students can be generalized, it is a matter of knowing how to *produce* what can thereafter be said to be there. Interpretation is not the art of construing but the art of constructing. Interpreters do not decode poems; they make them. (Fish 327)

The act of creating meaning from texts places interpretation and meaning not in an author to text to reader relationship, but in the act of reading itself, of which the reader has sole possession. The text appears not as something that conveys ideas *through* itself, but something the reader constructs meaning *out of*. This interesting perspective should not be overlooked, because it defines the methods of several contemporary poets.

As a Language poet and critic, Bruce Andrews argues that the traditional concept of meaning as an inherent

object--like a diamond in the earth--does not acknowledge its own indebtedness to use:

> Meaning isn't just a surplus value to be eliminated--It comes out of a productive *practice*. Not passively, as derivative of a system of differences (pre-defined) prior to composition. . . . Instead, active--back & forth: a relay constantly making contexts out of a fabric of markings: writing & reading. (Andrews 135)

The notion that meaning resides not in the text as object but in the text as process establishes a battlefield for contemporary poetry, even down to the level of the word itself. "The coherence between the signifier & signified is conventional, after all," Andrews notes, observing that "rather than skate past this fact, writing can rebel against it by breaking down that coherence, by negating the system itself" (134). Traditional poetry--what Jerome McGann calls "poetry of accommodation"--with its speaking self (or ego) and communicative function has a different agenda behind its construction, while some modern poetry (Language poetry included) operates with a politics that makes the reader aware of just how tenuous and arbitrary ink on the page can be (628). The variance between signifiers establishes their relation to each other, but Andrews explains that "writing can attack the structure of the sign after declaring that

settled system of differences to be repressive" (Andrews 134). Language, as a system of signs, allows us to interact with others and talk about objects around us, but repeated assaults on and reappropriations of the sign threaten this illusion of security. Andrews remarks that these attacks only reinforce our stubbornness to consider language as an effective instrument of communication:

> It's reached the point where a coercive organization of grammar, rhetoric, technical format & ideological symbols is normally imposed in everyday life to even get these eroded differences to do their job any more (an assembly line to deliver meaning, of certain kinds.) (Andrews 135)

Poets aware of this situation can exploit it, constructing a theory of poetics that McGann refers to as "oppositional" by exhibiting little concern about interpretation, "that positive obsession of academic discourse" and utilize nonsense, non-meaningful constructions, and fragmentation to achieve a level of "indetermination" (McGann 636). Through these tactics, Language writers try to "elucidate as it were the behavior, the manners, the way of life that various kinds of writings perform and live" (McGann 636).

Language poetry, in particular, works to counter preconceived notions of what a poem is or should be by

working outside of those definitions. Fish's students, to recall, operate with a set of assumptions that have been learned and reaffirmed in poetry classes. These strategies, right or wrong, influence the reader's construction of meaning, of which Fish explains:

> If your definition of poetry tells you that the language of poetry is complex, you will scrutinize the language of something identified as a poem in such a way to bring out the complexity you know to be "there." You will, for example, be on the look-out for latent ambiguities; you will attend to the presence of alliterative and consonal patterns (there will always be some), and you will try to make something of them (you will always succeed); you will search for meanings that subvert, or exist in a tension with the meanings that first present themselves; and if these operations fail to produce the anticipated complexity, you will even propose a significance for the words that are not there, because, as every one knows, everything about a poem, including its omissions, is significant. (Fish 327)

These criteria--a roadmap of reading strategies--help establish for "oppositional" poets a set of compositional

strategies to disrupt them. Since these poets, like Fish, view poetic discourse as not meaning-referential but meaning-constitutive, their work denies the "uncovering" of meaning in favor of the construction of it (McGann 636). While noting the diversity of the number of poets involved, McGann observes that Language poets "are involved with writing projects which fracture the surface regularities of the written text, and which interrupt conventional reading processes" (McGann 634).

Computer generated poetry could rightly be called a poetry of opposition because it, like Language poetry and other postmodern poetic forms and experiments, challenges readers' traditional assumptions of what constitutes and defines poetry. Analyzing the existing criteria some individuals have measured computer poetry against reveals subjective biases and problematic critical stances, yet the primary objection appears to be the loss of something that defines us as human. We should not, however, consider computer poetry as a replacement for human poets, but should view it instead as an impetus to reexamine our own poetic preferences. As Douglas Hofstadter acutely observes:

> What makes [computer generated poetry] seem reasonably convincing as poetry is mainly our cultural context: the fact that twentieth-century literature enormously widened the range of

> acceptability of poetry and prose. Our century's open-minded, "anything goes" attitude has definitely encouraged wonderful types of literary experimentation, but it has also made it far easier for impostors, human or otherwise, to crash the party and go completely undetected. (*Fluid Concepts* 470)

Yet, computers can only compose text; no computer or program can interpret and understand poetry. This may not pacify some, but the unique human ability to construct interpretations and to imagine what the scratches on the tablet or the ink on the page could represent seems to me a far more important area of study. Rather than bemoan the loss of a repressive system of signification, we should direct more attention to how interpretive communities shape and influence both our use and understanding of poetic language.

Notes:

1. Appendix 1 contains a copy of the actual survey, answers, and results of the those polled.

2. When the results were broken down into undergraduates, master's students or graduates, and doctorates, a slight increase in the number of correct answers appeared. Unfortunately, only 2 doctorates returned my survey, but the answers from those who did indicate that perhaps the level of education may affect one's recognition of computer poetry. However, the increase was only 13% between undergraduates and doctorates and only 3% between masters and doctorates. While doctorates scored the highest average of correct answers, this average of 56% hardly shows a considerable difference in aptitude.

3. For an interesting account of the postmodern attempt to make the machine invisible, see Strother B. Purdy, "Technopoetics: Seeing What Literature Has to Do with the Machine," in Critical Inquiry 11 (September 1984), pp. 130-40.

4. Obviously, one would have to be able to determine that the markings were made by the wind first, and not by human hands, in order to make this comparison. But suppose that some forms of erosion, like computer poetries, appear either

indistinguishable from human markings or that there exists no objective means for determining origin?

5. Beyond the scope of this paper is the question of what makes a "good" poem, or whether some poems exhibit more creativity than others. But an interesting response comes from Douglas Hofstadter, who considers the possibility of a certain computer application displaying creativity:

> Without some form of access to the innards of the program, I simply don't know how to evaluate the product, and so I can't decide if real creativity was involved or not. This attitude might seem strange to some people, who might say, 'What does it matter how it was made as long as it was made? A product is creative for certain external, objective reasons, not for how it came into existence!' But I don't feel that way. I cannot judge just the object before me; I feel a need to have a sense for its provenance, in some manner or other. (Fluid Concepts and Creative Analogies 481)

Conclusion

As my own survey shows, the sophistication of more recent computer generated poetry programs can create output that successfully mimics certain forms of human poetry. This troubles many individuals; in fact, while researching this work, I was approached by many aspiring poets who stated in no uncertain terms that "a computer cannot write poetry." Interestingly, the same persons who expressed such strong opinions could not determine decisively whether a human or computer generated the stanzas in my survey, with one respondent going so far as to write on the bottom of the form that none of the fifteen examples were poetry.

This last remark lays the groundwork for a debate about how we view poetry and how we consider the author's role in the production of such texts. The poststructural work of Barthes, Foucault, and Jacques Derrida, as well as the application of their ideas to poetic form appears to have very little appeal for people outside academia who may view poetry as a superlative form of communication or expression, a form of writing through which we, as humans, assert our emotions and our fears in the most dedicated and artful way. This elevation of the poem, accompanied by the responsibility and power associated with such a ennobling act, leads Shelley to his oft-quoted remark that poets "are the unacknowledged legislators of the world" (Shelley 969).

But the work of Russian Formalism, by attending to poetic form rather than origin; New Criticism, arguing the "intentional fallacy" of Wimsatt and Beardsley; and Poststructuralism, which asserts the play of signification and questions the boundaries between poetry and everyday speech; all distance or remove the poet from the work to some degree, leaving very little theoretical space through which to posit that "poetry" cannot exist without a "poet."

While I feel that the all-encompassing questions of "what is poetry" and "what role does the art perform" lie beyond the scope of this work because of the dynamic nature of poetry and theory and due to the difficulty in decisively determining boundaries for the art, this study of computer generated poetry does work to reveal several preconceived biases about poetry and theory, particularly in relation to established notions of language as a strictly human domain. Even those poetic movements (such as Dadaism, Surrealism, Modernism, and Language Writing) that react in opposition to dominant traditions, always reaffirm a human presence, (at the very least, the original presence of a "rational" or "intelligent" being) even if they take such presence for granted. The level of technology emerging today reveals that we can no longer assume a human author for some forms of poetry; text generating programs, rather than reveal the

person behind Oz, reinforce the curtain that separates the mode of production from the product itself.

At this stage of computer development, our claims about language and the desires we place on language are not threatened. As I note in Chapter three, the computer can only mimic *certain kinds* of poetry, and certain structures lie outside of the reach of today's efforts in programming. Genre forms such as epic, dramatic lyric, sonnet, etc., are difficult if not impossible for a computer to generate--although very few human poets today write in these forms. Ironically, contemporary poetry, with its emphasis on free verse form and its frequent avoidance of the lyric qualities of pre-modern and modern verse advanced the landscape of poetic form to a point where it converges with computer technology. By positing the absent "subject" or reducing authorship to a "function," poststructural theory coincides with developments in poetry that attempt to unwork the presence or centering of the speaking subject; Foucault cautions that the subject "is not in fact the cause, origin, or starting-point of the phenomenon of the written or spoken articulation of the sentence" (*Archaeology* 95). The "I" or ego of the text becomes disallowed because no one "owns" language.

The intrusion of the computer further complicates the notion of authorship because through aleatory functions,

multiple input texts, and the indistinguishable contribution of those people involved, the computer reveals a conventional prejudice for the "who" behind the text as opposed to "what." What is at stake in any serious discussion of computer generated poetry is not the loss of our humanness, but how we have traditionally considered the origin of the text as crucial. In the worst case, the concept of authorial "intention" can represent inflexibility, repression, and inaccessibility because the work becomes a one-way conduit for transmission of meaning and idea. On the other hand, postructuralism and oppositional poetry (including Language poetry) attempt to shift critical attention away from the producer of the text, focusing instead on the reception of the text and questioning the ways in which readers have traditionally read poetry. Computer generated poetry contributes to this debate by placing emphasis on the way poetry and all art forms serve as a *catalyst* for human thought, reducing our need to concentrate on either the textual form or the author. As readers in an interpretive community, we may need to shift our discourse to a concern about what we do with poetry, rather than concentrate on where poetry comes from or what poetry does, because how we read any text still depends on our uniquely human condition.

APPENDIX:
COMPUTER POETRY SURVEY AND RESULTS
A. SURVEY

DO YOU READ POETRY: ALMOST NEVER OCCASIONALLY IT IS MY LIFE

LEVEL OF EDUCATION: UNDERGRAD MASTER'S DOCTORATE

1. HUMAN or COMPUTER
The inner sun
attempts summer stillness
O cloistered ineffable!

2. HUMAN or COMPUTER
Sentences for Analysis and Parsing Thayer Street
Grammar School begins. James, bring me the
vessel had been using that that. Our little lame.
He hurricane. The love of money is to prepare
forsaken. Iron has brought it tremble. The young
must do it is.

3. HUMAN or COMPUTER
He had a stroke of luck
where beasts lick their paws
of your armchairs and the fortune
cookie right eye of your surprising
spectacles carries the word
like Typhoid Mary, dragging bones
through green felt enough
that rien ne va plus
His last words, "Utah Shale and Advanced Ross,"
smile where bubbles burst.

4. HUMAN or COMPUTER
An angry writer remorselessly writes
about an old glass
A poem deliberately mangles shoes.

Walking--
A woman sweeps the blue and violet skies
Dissonantly felt, because people listen sexually
A verdant devil abruptly drives over loneliness.
An angry watch
Mirrors a city, and further
A decayed feeling about mirrors sings
A thought that rarely comes on poems.

5. HUMAN or COMPUTER
But could it come up into a limestone so correct, teeth would be slim by comparison. Have a go under the waterfall for health and a mouth to pour. White powder pile could be of snow or rock in flake. Seeds that hold all lime in ledge to grasp.

6. HUMAN or COMPUTER
That force is lost
which shaped me, spent
in its image, battered, an old brown thing
swept off the streets
where it sucked its
gentle living.
And what is meat
to do, that is driven to its end
by words? The frailest gestures
grown like skirts around breathing.
We take
unholy risks to prove
we are what we cannot be. For instance,

I am not even crazy.

7. HUMAN or COMPUTER
In the network, in the ruin,
flashing classics gravitate,
snared, encumbered voicelessly.

Teak enticements seek, leaping
fan-shaped arras corners
snore among in backward dispatch.

Panels glow, groan, territorialize
fetishistically in nacreous
instantaneity spookily shod.

8. HUMAN or COMPUTER
Red river, red river,
Slow flow heat is silence
No will is still as a river
Still. Will heat move
Only through the morning-bird
Heard once? Still hills
Wait. Gates wait. Purple trees,
white trees, wait, wait,
Delay. decay. Living, living,

Never moving. Ever moving
Iron thoughts came with me
And go with me:
Red river, river, river.

9. HUMAN or COMPUTER
O thou,
Who moved among some fierce Maenad, even among noise
and blue
Between the bones sang, scattered and the silent seas.

10. HUMAN or COMPUTER
Derrida loves awhile, he understands absolutely and ritually
his pert fad
Nature accepts from him
these bombshells like
a number of fatalities.
They corrode,
I examine fantastic optimistic mysterious losses.
Derrida suffers for us
his esoteric want stirs
a lot of profanities,
Why are eloquent unrealistic fictions
like damned societies?
because fictions confide timidly.
Derrida begs for me,
my soul is like his fuzzy passivity.

11. HUMAN or COMPUTER
From the dying pastoral slopes an unwanted earth art gone
And the vast edges draw back the impulse of an hour--
Exhausted, thou waitest for one desire, and the soft
Abstractions of reapers in the intellectual trough.
So wild brother men, concealed then with distracted air--
Let it be spent on other joy, and we,
Wanderer one of antique shadow, rest
And in the bluebell-drenched days, men
Who in the sun, thy fire their being roll.
Come shepard, bathe in our war of antique shadow,
'Tis this story of the wooden bridge, wrapt in disguise.

12. HUMAN or COMPUTER
The cleat curved you curved the spider
the coil of alcoholic fumes
the webbing of sail & sunset,

over the mountain the distance: Colorado,
New Mexico. In Tucson
the beggars are gymnasts good riders swaying side-to-side
are steerers covering much territory
the backroom towel & soap the front leg.

13. HUMAN or COMPUTER
comfort notions
   correction
incapable of keeping
   case-histories

foresee requisites
   talk about
a grinning idiot

   (closed down
nothing comes up
they ram a car
   up there
instead of a cop)

14. HUMAN or COMPUTER
Nature stimulates she quivers
          with affections,
Nature detests
     your own orthodox, ambitious ideas from Heaven
Her captivation requires dreams
     no atrocity likes
          her brinkmanship:
both cooperate allegedly or symbolically while
     a single bold actuality strips and
          she heaves her captor
her publicity is like her debt:
     it sings to universe.
Nature longs for the poet
her asylum requires this black flame.

15. HUMAN or COMPUTER
The bird covets her own victory;
Then guesses the company;
In her silent truth buzz no more.

The definition presumes her own thing;
Then covets the victory;
Of her condensed journey buzz no more.

The thing presumes her own civility;
then advocates the nectar;
With her forbidden victory perish no more.

B. ANSWERS:

1. Generated by *Haiku Master*.

2. Generated by Hugh Kenner's *Travesty* given an input text consisting of an elementary school grammar workbook from the late 1800s.

3. "No Chance Operations" by James Sherry.

4. Generated by *DadaPoem Generator v.1.0* (I have modified the program).

5. From "Manlius to Coeymans" by Clark Coolidge.

6. "Snake Eyes" by Amiri Baraka (LeRoi Jones).

7. "Trope Market" by Jackson Mac Low.

8. "Virginia" from *Landscapes* by T. S. Eliot.

9. Generated by the *Kurzweil Computer Poet* utilizing an input text consisting of poems from both P. B. Shelley and T. S. Eliot.

10. Generated by *McPoet*. Before running, I input "Derrida" as a male-gendered subject. The line breaks are somewhat random.

11. Generated by *Brekdown* after analyzing an input text consisting of Matthew Arnold's "The Buried Life," "Dover Beach," and "The Scholar-Gypsy." Edited to an unknown extent by John Tranter.

12. From "Lapstrake" by Ted Greenwald.

13. "Road" by Kevin Magee.

14. Generated by *McPoet*. I input "Nature" as a female-gendered subject.

15. Generated by *The Poetry Generator,* after analyzing an input text consisting of Emily Dickinson's poetry. Grammatical errors edited out by George Stewart.

C. RESULTS OF THE HUMAN/COMPUTER POET SURVEY:

PERCENTAGE OF CORRECT ANSWERS BY CATEGORY OF EDUCATION:

Question #:	UNDERGRAD (26)	MASTER'S (9)	DOCTORATE (2)	OVERALL (37)
1	42%	55%	50%	46%
2	88%	64%	50%	79%
3	62%	45%	50%	56%
4	15%	45%	50%	26%
5	46%	27%	50%	41%
6	88%	91%	100%	90%
7	23%	64%	100%	38%
8	58%	91%	50%	67%
9	38%	36%	50%	38%
10	27%	55%	0%	33%
11	46%	18%	100%	36%
12	35%	45%	50%	38%
13	19%	45%	50%	28%
14	19%	27%	0%	21%
15	38%	91%	100%	56%
AVERAGE:	43%	53%	56%	46.2%

Works Cited

Andrews, Bruce. "Writing Social Work & Political Practice." The L=A=N=G=U=A=G=E Book. Ed. Bruce Andrews and Charles Bernstein. Carbondale, IL: Southern Illinois UP, 1984.

Barthes, Roland. "The Death of the Author." Contemporary Literary Theory. Ed. Dan Latimer. New York: Harcourt, 1989. 54-59.

Bernstein, Charles. Content's Dream: Essays 1975-1984. Los Angeles: Sun & Moon, 1986.

Barger, Jorn. "'The Policeman's Beard' Was Largely Prefab!" Journal of Computer Game Design. August 1993.

Bond, Donald F., ed. The Spectator. London: Oxford UP, 1965.

Borges, Jorge Luis. Labyrinths: Selected Stories and Other Writings. New York: New Directions, 1962.

Chachanashvili, Alex. DadaPoem Generator. version 1.0. 1991.

Chamberlain, William. The Policeman's Beard is Half-Constructed. New York: Warner, 1984.

Conte, Joseph M. Unending Design: The Forms of Postmodern Poetry. Ithaca, NY: Cornell UP, 1991.

Davidson, Donald. "What Metaphors Mean." On Metaphor. Ed. Sheldon Sacks. Chicago: University of Chicago Press, 1979.

Dreyfus, Hubert. What Computers Still Can't Do: A Critique of Artificial Reason. Cambridge: MIT, 1994.

Erickson, John D. Dada: Performance, Poetry, and Art. Boston: Twayne, 1984.

Ernst, Josef. "Computer Poetry: An Act of Disinterested Communication." New Literary History. 23(2) (1992): 451-65.

Fish, Stanley. Is There a Text in This Class? Cambridge: Harvard UP, 1980.

Foucault, Michel. The Archaeology of Knowledge. Trans. A. M. Sheridan Smith. New York: Pantheon, 1972.

---. "What is an Author?" Textual Strategies. Ed. Jose V. Harari. Ithaca, NY: Cornell UP, 1979. 262-275.

Hartman, Charles and Hugh Kenner. Sentences. Los Angeles: Sun and Moon, 1995.

Hofstadter, Douglas. Fluid Concepts and Creative Analogies. New York: Basic, 1995.

---. Gödel, Escher, Bach. New York: Vintage, 1980.

Hugnet, Georges. "The Dada Spirit in Painting. " The Dada Painters and Poets: An Anthology. Ed. Robert Motherwell. New York: Wittenborn and Schultz, 1951.

Juhl, P. D. "Do Computer Poems Show That an Author's Intention Is Irrelevant to the Meaning of a Literary Work?" Critical Inquiry. (1979): 481-87.

Kurzweil, Ray. The Age of Intelligent Machines. Cambridge,

MA: MIT, 1990.

Mac Low, Jackson. Letter to Karin Schalm. 23 November 1994.

---. The Virginia Woolf Poems. Providence, RI: Burning Deck, 1985.

Masterman, Margaret and Robin McKinnon Wood. "Computerized Japanese Haiku." Cybernetic Serendipity. Ed. Jasia Reichardt. New York: Frederick A. Praeger, 1969.

McGann, Jerome. "Contemporary Poetry, Alternate Routes." Critical Inquiry. 13 (1987): 624-47.

Milic, Louis T. "The Possible Usefulness of Poetry Generation." The Computer in Literary and Linguistic Research. Ed. R. A. Wisbey. London: Cambridge UP, 1971.

Nemerov, Howard. "Speculative Equations: Poems, Poets, Computers." American Scholar. 16 (1967): 394-414.

Newell, Kenneth B. "Pattern, Concrete, and Computer Poetry: The Poem as Object in Itself." Bucknell Review: A Scholarly Journal of Letters, Arts and Sciences 27. 2 (1983): 159-73.

Nicholson, Marjorie. Science and Imagination. Ithaca, NY: Great Seal Books, 1956.

Perloff, Marjorie. "The Word as Such: L=A=N=G=U=A=G=E Poetry in the Eighties." American Poetry Review. 13 (May/June 1984): 15-22.

Peter, John. Artificial Versifying: A New Way to Make Latin Verses. 2nd. Ed. London: By T. J. for John Sims, 1678.

Richter, Hans. DADA: Art and Anti-Art. New York: McGraw-Hill, 1964.

Saussure, Ferdinand de. "Course in General Linguistics." The Structuralists from Marx to Levi-Strauss. Ed. Richard and Fernande De George. Garden City, NY: Anchor, 1972. 59-79.

Shelley, Percy Bysshe. "A Defence of Poetry; or, Remarks Suggested by an Essay Entitled 'The Four Ages of Poetry'." Romanticism: An Anthology. Ed. Duncan Wu. Oxford: Blackwell, 1994. 956-69.

Sincoff, Erik. Poetry CreatOR.

Stone, Andrew. Haiku Master. Version 2.2. 1988. Modified by Brian Thomas, 1989.

Strack, Stefan. Mark V. Shaney. Version 1.0. 1991.

Swift, Jonathan. Gulliver's Travels. New York: W. W. Norton, 1970.

Tranter, John. "Mr. Rubenking's 'Brekdown'." Postmodern Culture 3.1 (Sept., 1992): n. pag. Online. Internet. 15 November 1995.

Tzara, Tristan. Seven Dada Manifestos and Lampisteries. Trans. Barbara Wright. New York: Riverrun, 1992.

Watten, Barrett. Total Syntax. Carbondale, IN: Southern

Illinois UP, 1985.

Watts, Harriet Ann. Chance: A Perspective on Dada. Ann Arbor, Michigan: UMI Research, 1975.

Wellek, René. "The Analysis of the Literary Work of Art." Modern Criticism: Theory and Practice. Ed. Walter Sutton and Richard Foster. New York: Odyssey, 1963. 257-261.

Westbury, Chris. McPoet. Version 2.0. 1994.

Wimsatt, W. K. and Monroe Beardsley. "The Intentional Fallacy." Modern Criticism: Theory and Practice. Ed. Walter Sutton and Richard Foster. New York: Odyssey, 1963. 248-257.

Wittgenstein, Ludwig. Philosophical Investigations. 3rd ed. New York: Macmillan, 1958.

---. The Blue and the Brown Books. 2nd ed. New York: Harper & Row, 1960.